

party Reference Manual

Generated by Doxygen 1.4.6

Mon Jul 23 13:17:14 2007

Contents

Chapter 1

party Directory Hierarchy

1.1 party Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

src ??

Chapter 2

party File Index

2.1 party File List

Here is a list of all files with brief descriptions:

Classes.c	??
Classes.h	??
Convenience.c	??
Convenience.h	??
Distributions.c	??
Distributions.h	??
IndependenceTest.c	??
IndependenceTest.h	??
LinearStatistic.c	??
LinearStatistic.h	??
mvt.h	??
Node.c	??
Node.h	??
party.h	??
Predict.c	??
Predict.h	??
RandomForest.c	??
S3Classes.c	??
S3Classes.h	??
Splits.c	??
Splits.h	??
SurrogateSplits.c	??
SurrogateSplits.h	??
TestStatistic.c	??
TestStatistic.h	??
TreeGrow.c	??
TreeGrow.h	??
Utils.c	??
Utils.h	??

Chapter 3

party Page Index

3.1 party Related Pages

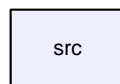
Here is a list of all related documentation pages:

Todo List	??
---------------------	----

Chapter 4

party Directory Documentation

4.1 src/ Directory Reference



Files

- file [Classes.c](#)
- file [Classes.h](#)
- file [Convenience.c](#)
- file [Convenience.h](#)
- file [Distributions.c](#)
- file [Distributions.h](#)
- file [IndependenceTest.c](#)
- file [IndependenceTest.h](#)
- file [LinearStatistic.c](#)
- file [LinearStatistic.h](#)
- file [mvt.h](#)
- file [Node.c](#)
- file [Node.h](#)
- file [party.h](#)
- file [Predict.c](#)
- file [Predict.h](#)
- file [RandomForest.c](#)
- file [S3Classes.c](#)
- file [S3Classes.h](#)
- file [Splits.c](#)
- file [Splits.h](#)
- file [SurrogateSplits.c](#)
- file [SurrogateSplits.h](#)
- file [TestStatistic.c](#)
- file [TestStatistic.h](#)

- file [TreeGrow.c](#)
- file [TreeGrow.h](#)
- file [Utils.c](#)
- file [Utils.h](#)

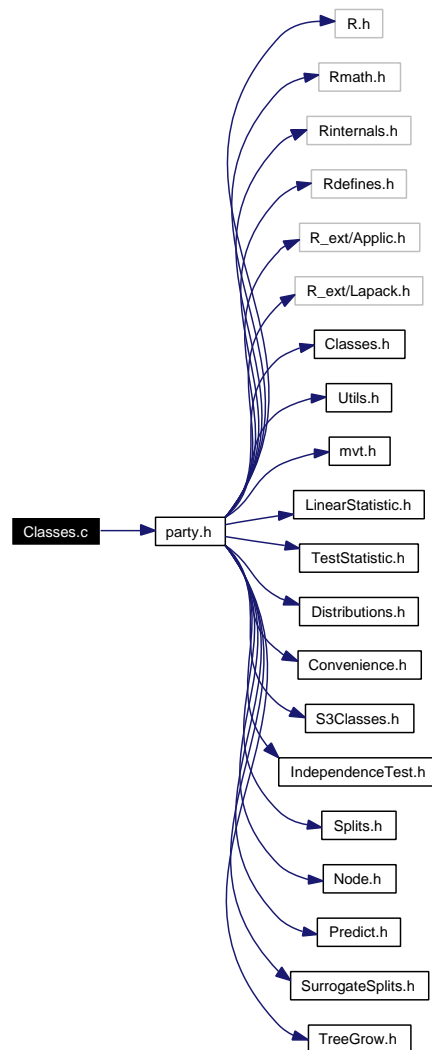
Chapter 5

party File Documentation

5.1 Classes.c File Reference

```
#include "party.h"
```

Include dependency graph for Classes.c:



Functions

- SEXP [party_init](#) (void)
- int [get_dimension](#) (SEXP object)
- int [get_teststat](#) (SEXP object)
- int [get_pvalue](#) (SEXP object)
- double [get_tol](#) (SEXP object)
- int [get_maxpts](#) (SEXP object)
- double [get_abseps](#) (SEXP object)
- double [get_releps](#) (SEXP object)
- double [get_minsplit](#) (SEXP object)
- double [get_minprob](#) (SEXP object)
- double [get_minbucket](#) (SEXP object)
- SEXP [get_transformation](#) (SEXP object, int variable)
- SEXP [get_test_trafo](#) (SEXP object)
- SEXP [get_predict_trafo](#) (SEXP object)
- SEXP [get_variable](#) (SEXP object, int variable)

- int [is_nominal](#) (SEXP object, int variable)
- int [is_ordinal](#) (SEXP object, int variable)
- int [is_censored](#) (SEXP object, int variable)
- int [has_missings](#) (SEXP object, int variable)
- SEXP [get_ordering](#) (SEXP object, int variable)
- SEXP [get_levels](#) (SEXP object, int variable)
- SEXP [get_scores](#) (SEXP object, int variable)
- SEXP [get_missings](#) (SEXP object, int variable)
- SEXP [get_varmemory](#) (SEXP object, int variable)
- int [get_savesplitstats](#) (SEXP object)
- SEXP [get_splitstatistics](#) (SEXP object)
- int [get_nobs](#) (SEXP object)
- int [get_ninputs](#) (SEXP object)
- SEXP [get_weights](#) (SEXP object, int variable)
- int [get_testtype](#) (SEXP object)
- int [get_nresample](#) (SEXP object)
- SEXP [get_varctrl](#) (SEXP object)
- SEXP [get_splitctrl](#) (SEXP object)
- SEXP [get_gtctrl](#) (SEXP object)
- SEXP [get_tgctrl](#) (SEXP object)
- double [get_mincriterion](#) (SEXP object)
- int [get_maxsurrogate](#) (SEXP object)
- int [get_randomsplits](#) (SEXP object)
- int [get_mtry](#) (SEXP object)
- SEXP [get_dontuse](#) (SEXP object)
- SEXP [get_dontusetmp](#) (SEXP object)
- int [get_stump](#) (SEXP object)
- int [check_depth](#) (SEXP object, int depth)
- int [get_ntree](#) (SEXP object)
- int [get_replace](#) (SEXP object)
- double [get_fraction](#) (SEXP object)

Variables

- SEXP [PL2_expectationSym](#)
- SEXP [PL2_covarianceSym](#)
- SEXP [PL2_linearstatisticSym](#)
- SEXP [PL2_expcovinfSym](#)
- SEXP [PL2_expcovinfssSym](#)
- SEXP [PL2_sumweightsSym](#)
- SEXP [PL2_dimensionSym](#)
- SEXP [PL2_MPinvSym](#)
- SEXP [PL2_rankSym](#)
- SEXP [PL2_svdmemSym](#)
- SEXP [PL2_methodSym](#)
- SEXP [PL2_jobuSym](#)
- SEXP [PL2_jobvSym](#)
- SEXP [PL2_uSym](#)
- SEXP [PL2_vSym](#)
- SEXP [PL2_sSym](#)

- [SEXP PL2_pSym](#)
- [SEXP PL2_teststatSym](#)
- [SEXP PL2_pvalueSym](#)
- [SEXP PL2_tolSym](#)
- [SEXP PL2_maxptsSym](#)
- [SEXP PL2_absepsSym](#)
- [SEXP PL2_relepsSym](#)
- [SEXP PL2_minprobSym](#)
- [SEXP PL2_minsplitSym](#)
- [SEXP PL2_minbucketSym](#)
- [SEXP PL2_variablesSym](#)
- [SEXP PL2_transformationsSym](#)
- [SEXP PL2_is_nominalSym](#)
- [SEXP PL2_is_ordinalSym](#)
- [SEXP PL2_is_censoredSym](#)
- [SEXP PL2_orderingSym](#)
- [SEXP PL2_levelsSym](#)
- [SEXP PL2_scoresSym](#)
- [SEXP PL2_has_missingsSym](#)
- [SEXP PL2_whichNASym](#)
- [SEXP PL2_test_trafoSym](#)
- [SEXP PL2_predict_trafoSym](#)
- [SEXP PL2_nobsSym](#)
- [SEXP PL2_ninputsSym](#)
- [SEXP PL2_linexpcov2sampleSym](#)
- [SEXP PL2_weightsSym](#)
- [SEXP PL2_varmemorySym](#)
- [SEXP PL2_splitstatisticsSym](#)
- [SEXP PL2_savesplitstatsSym](#)
- [SEXP PL2_responsesSym](#)
- [SEXP PL2_inputsSym](#)
- [SEXP PL2_testtypeSym](#)
- [SEXP PL2_nresampleSym](#)
- [SEXP PL2_varctrlSym](#)
- [SEXP PL2_splitctrlSym](#)
- [SEXP PL2_gtctrlSym](#)
- [SEXP PL2_mincriterionSym](#)
- [SEXP PL2_maxsurrogateSym](#)
- [SEXP PL2_randomsplitsSym](#)
- [SEXP PL2_mtrySym](#)
- [SEXP PL2_dontuseSym](#)
- [SEXP PL2_dontusetmpSym](#)
- [SEXP PL2_stumpSym](#)
- [SEXP PL2_maxdepthSym](#)
- [SEXP PL2_tgctrlSym](#)
- [SEXP PL2_ntreeSym](#)
- [SEXP PL2_replaceSym](#)
- [SEXP PL2_fractionSym](#)

5.1.1 Detailed Description

S4 classes for package ‘party’

Author

hothorn

Date

2007-06-20 18:11:19 +0200 (Wed, 20 Jun 2007)

Definition in file [Classes.c](#).

5.1.2 Function Documentation

5.1.2.1 `int check_depth (SEXP object, int depth)`

Definition at line 346 of file [Classes.c](#).

References [PL2_maxdepthSym](#).

Referenced by [C_TreeGrow\(\)](#).

5.1.2.2 `double get_abseps (SEXP object)`

Definition at line 167 of file [Classes.c](#).

References [PL2_absepsSym](#).

Referenced by [C_TeststatPvalue\(\)](#).

5.1.2.3 `int get_dimension (SEXP object)`

Definition at line 147 of file [Classes.c](#).

References [PL2_dimensionSym](#).

Referenced by [C_ConditionalPvalue\(\)](#), [C_Node\(\)](#), [C_TestStatistic\(\)](#), and [R_splitcategorical\(\)](#).

5.1.2.4 `SEXP get_dontuse (SEXP object)`

Definition at line 334 of file [Classes.c](#).

References [PL2_dontuseSym](#).

5.1.2.5 `SEXP get_dontusetmp (SEXP object)`

Definition at line 338 of file [Classes.c](#).

References [PL2_dontusetmpSym](#).

5.1.2.6 `double get_fraction (SEXP object)`

Definition at line 362 of file [Classes.c](#).

References [PL2_fractionSym](#).

5.1.2.7 SEXP get_gtctrl (SEXP object)

Definition at line 310 of file Classes.c.

References PL2_gtctrlSym.

Referenced by C_Node().

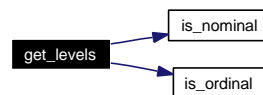
5.1.2.8 SEXP get_levels (SEXP object, int variable)

Definition at line 235 of file Classes.c.

References is_nominal(), is_ordinal(), and PL2_levelsSym.

Referenced by C_Node().

Here is the call graph for this function:

**5.1.2.9 int get_maxpts (SEXP object)**

Definition at line 163 of file Classes.c.

References PL2_maxptsSym.

Referenced by C_TeststatPvalue().

5.1.2.10 int get_maxsurrogate (SEXP object)

Definition at line 322 of file Classes.c.

References PL2_maxsurrogateSym.

Referenced by C_splitnode(), C_surrogates(), C_TreeGrow(), R_Node(), and R_TreeGrow().

5.1.2.11 double get_minbucket (SEXP object)

Definition at line 183 of file Classes.c.

References PL2_minbucketSym.

5.1.2.12 double get_mincriterion (SEXP object)

Definition at line 318 of file Classes.c.

References PL2_mincriterionSym.

Referenced by C_Node().

5.1.2.13 double get_minprob (SEXP object)

Definition at line 179 of file Classes.c.

References PL2_minprobSym.

5.1.2.14 double get_minsplit (SEXP object)

Definition at line 175 of file Classes.c.

References PL2_minsplitSym.

Referenced by C_Node().

5.1.2.15 SEXP get_missings (SEXP object, int variable)

Definition at line 258 of file Classes.c.

References has_missings(), and PL2_whichNASym.

Referenced by C_get_node(), C_splitnode(), C_splitsurrogate(), and C_surrogates().

Here is the call graph for this function:

**5.1.2.16 int get_mtry (SEXP object)**

Definition at line 330 of file Classes.c.

References PL2_mtrySym.

5.1.2.17 int get_ninputs (SEXP object)

Definition at line 286 of file Classes.c.

References PL2_ninputsSym.

Referenced by C_GlobalTest(), C_MonteCarlo(), C_Node(), C_splitnode(), C_surrogates(), R_GlobalTest(), R_MonteCarlo(), R_Node(), and R_TreeGrow().

5.1.2.18 int get_nobs (SEXP object)

Definition at line 282 of file Classes.c.

References PL2_nobsSym.

Referenced by C_GlobalTest(), C_MonteCarlo(), C_Node(), C_predict(), C_splitnode(), C_splitsurrogate(), C_surrogates(), C_TreeGrow(), R_Ensemble(), R_get_nodeID(), R_Node(), R_predict(), R_predictRF_weights(), and R_TreeGrow().

5.1.2.19 int get_nresample (SEXP *object*)

Definition at line 298 of file Classes.c.

References PL2_nresampleSym.

Referenced by C_MonteCarlo().

5.1.2.20 int get_ntree (SEXP *object*)

Definition at line 354 of file Classes.c.

References PL2_ntreeSym.

Referenced by R_Ensemble().

5.1.2.21 SEXP get_ordering (SEXP *object*, int *variable*)

Definition at line 224 of file Classes.c.

References is_nominal(), and PL2_orderingSym.

Referenced by C_Node().

Here is the call graph for this function:

**5.1.2.22 SEXP get_predict_trafo (SEXP *object*)**

Definition at line 197 of file Classes.c.

References PL2_predict_trafoSym.

Referenced by C_Node(), C_splitnode(), R_modify_response(), R_Node(), R_set_response(), and R_TreeGrow().

5.1.2.23 int get_pvalue (SEXP *object*)

Definition at line 155 of file Classes.c.

References PL2_pvalueSym.

Referenced by C_TeststatCriterion(), and C_TeststatPvalue().

5.1.2.24 int get_randomsplits (SEXP *object*)

Definition at line 326 of file Classes.c.

References PL2_randomsplitsSym.

5.1.2.25 double get_releps (SEXP *object*)

Definition at line 171 of file Classes.c.

References PL2_relepsSym.

Referenced by C_TeststatPvalue().

5.1.2.26 int get_replace (SEXP object)

Definition at line 358 of file Classes.c.

References PL2_replaceSym.

5.1.2.27 int get_savesplitstats (SEXP object)

Definition at line 274 of file Classes.c.

References PL2_savesplitstatsSym.

Referenced by C_Node().

5.1.2.28 SEXP get_scores (SEXP object, int variable)

Definition at line 247 of file Classes.c.

References is_ordinal(), and PL2_scoresSym.

Here is the call graph for this function:



5.1.2.29 SEXP get_splitctrl (SEXP object)

Definition at line 306 of file Classes.c.

References PL2_splitctrlSym.

Referenced by C_Node(), C_splitnode(), C_surrogates(), C_TreeGrow(), R_Node(), and R_TreeGrow().

5.1.2.30 SEXP get_splitstatistics (SEXP object)

Definition at line 278 of file Classes.c.

References PL2_splitstatisticsSym.

Referenced by C_Node().

5.1.2.31 int get_stump (SEXP object)

Definition at line 342 of file Classes.c.

References PL2_stumpSym.

Referenced by C_TreeGrow().

5.1.2.32 SEXP get_test_trafo (SEXP object)

Definition at line 193 of file Classes.c.

References PL2_test_trafoSym.

Referenced by C_GlobalTest(), C_MonteCarlo(), C_Node(), R_modify_response(), and R_set_response().

5.1.2.33 int get_teststat (SEXP object)

Definition at line 151 of file Classes.c.

References PL2_teststatSym.

Referenced by C_IndependenceTest(), and C_TeststatPvalue().

5.1.2.34 int get_testtype (SEXP object)

Definition at line 294 of file Classes.c.

References PL2_testtypeSym.

5.1.2.35 SEXP get_tgctrl (SEXP object)

Definition at line 314 of file Classes.c.

References PL2_tgctrlSym.

Referenced by C_Node(), and C_TreeGrow().

5.1.2.36 double get_tol (SEXP object)

Definition at line 159 of file Classes.c.

References PL2_tolSym.

Referenced by C_IndependenceTest(), C_Node(), C_split(), C_splitcategorical(), C_TeststatPvalue(), and R_splitcategorical().

5.1.2.37 SEXP get_transformation (SEXP object, int variable)

Definition at line 187 of file Classes.c.

References PL2_transformationsSym.

Referenced by C_Node(), and R_modify_response().

5.1.2.38 SEXP get_varctrl (SEXP object)

Definition at line 302 of file Classes.c.

References PL2_varctrlSym.

Referenced by C_Node().

5.1.2.39 SEXP get_variable (SEXP object, int variable)

Definition at line 202 of file Classes.c.

References PL2_variablesSym.

Referenced by C_get_node(), C_Node(), C_splitnode(), C_splitsurrogate(), and R_modify_response().

5.1.2.40 SEXP get_varmemory (SEXP object, int variable)

Definition at line 269 of file Classes.c.

References PL2_varmemorySym.

Referenced by C_Node().

5.1.2.41 SEXP get_weights (SEXP object, int variable)

Definition at line 290 of file Classes.c.

References PL2_weightsSym.

Referenced by C_Node().

5.1.2.42 int has_missings (SEXP object, int variable)

Definition at line 220 of file Classes.c.

References PL2_has_missingsSym.

Referenced by C_get_node(), C_Node(), C_splitnode(), C_splitsurrogate(), C_surrogates(), and get_missings().

5.1.2.43 int is_censored (SEXP object, int variable)

Definition at line 216 of file Classes.c.

References PL2_is_censoredSym.

5.1.2.44 int is_nominal (SEXP object, int variable)

Definition at line 208 of file Classes.c.

References PL2_is_nominalSym.

Referenced by C_Node(), get_levels(), and get_ordering().

5.1.2.45 int is_ordinal (SEXP object, int variable)

Definition at line 212 of file Classes.c.

References PL2_is_ordinalSym.

Referenced by get_levels(), and get_scores().

5.1.2.46 SEXP party_init (void)

Definition at line 77 of file Classes.c.

References PL2_absepsSym, PL2_covarianceSym, PL2_dimensionSym, PL2_dontuseSym, PL2_dontusetmpSym, PL2_expcovinfssSym, PL2_expcovinfSym, PL2_expectationSym, PL2_fractionSym, PL2_gtctrlSym, PL2_has_missingsSym, PL2_inputsSym, PL2_is_censoredSym, PL2_is_nominalSym, PL2_is_ordinalSym, PL2_jobuSym, PL2_jobvSym, PL2_levelsSym, PL2_linearstatisticSym, PL2_linexpcov2sampleSym, PL2_maxdepthSym, PL2_maxptsSym, PL2_maxsurrogateSym, PL2_methodSym, PL2_minbucketSym, PL2_mincriterionSym, PL2_minprobSym, PL2_minsplitSym, PL2_MPinvSym, PL2_mtrySym, PL2_ninputsSym, PL2_nobsSym, PL2_nresampleSym, PL2_ntreeSym, PL2_orderingSym, PL2_predict_trafoSym, PL2_pSym, PL2_pvalueSym, PL2_randsplitsSym, PL2_rankSym, PL2_relepsSym, PL2_replaceSym, PL2_responsesSym, PL2_savesplitstatsSym, PL2_scoresSym, PL2_splitctrlSym, PL2_splitstatisticsSym, PL2_sSym, PL2_stumpSym, PL2_sumweightsSym, PL2_svdmemSym, PL2_test_trafoSym, PL2_teststatSym, PL2_testtypeSym, PL2_tgctrlSym, PL2_tolSym, PL2_transformationsSym, PL2_uSym, PL2_varctrlSym, PL2_variablesSym, PL2_varmemorySym, PL2_vSym, PL2_weightsSym, and PL2_whichNASym.

5.1.3 Variable Documentation

5.1.3.1 SEXP PL2_absepsSym

Definition at line 12 of file Classes.c.

Referenced by get_abseps(), and party_init().

5.1.3.2 SEXP PL2_covarianceSym

Definition at line 12 of file Classes.c.

Referenced by C_ConditionalPvalue(), C_ExpectCovarInfluence(), C_ExpectCovarLinearStatistic(), C_LinStatExpCovMPinv(), C_Node(), C_TestStatistic(), party_init(), R_ExpectCovarInfluence(), R_ExpectCovarLinearStatistic(), and R_splitcategorical().

5.1.3.3 SEXP PL2_dimensionSym

Definition at line 12 of file Classes.c.

Referenced by get_dimension(), and party_init().

5.1.3.4 SEXP PL2_dontuseSym

Definition at line 12 of file Classes.c.

Referenced by get_dontuse(), and party_init().

5.1.3.5 SEXP PL2_dontusetmpSym

Definition at line 12 of file Classes.c.

Referenced by get_dontusetmp(), and party_init().

5.1.3.6 SEXP [PL2_expcovinfssSym](#)

Definition at line 12 of file Classes.c.

Referenced by `party_init()`.

5.1.3.7 SEXP [PL2_expcovinfSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_GlobalTest()`, `C_IndependenceTest()`, `C_MonteCarlo()`, `C_Node()`, `party_init()`, and `R_splitcategorical()`.

5.1.3.8 SEXP [PL2_expectationSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_ExpectCovarInfluence()`, `C_ExpectCovarLinearStatistic()`, `C_Node()`, `C_TestStatistic()`, `party_init()`, `R_ExpectCovarInfluence()`, `R_ExpectCovarLinearStatistic()`, and `R_splitcategorical()`.

5.1.3.9 SEXP [PL2_fractionSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_fraction()`, and `party_init()`.

5.1.3.10 SEXP [PL2_gtctrlSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_gtctrl()`, and `party_init()`.

5.1.3.11 SEXP [PL2_has_missingsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `has_missings()`, and `party_init()`.

5.1.3.12 SEXP [PL2_inputsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_GlobalTest()`, `C_MonteCarlo()`, `C_Node()`, `C_splitnode()`, `C_splitsurrogate()`, `C_surrogates()`, and `party_init()`.

5.1.3.13 SEXP [PL2_is_censoredSym](#)

Definition at line 12 of file Classes.c.

Referenced by `is_censored()`, and `party_init()`.

5.1.3.14 SEXP [PL2_is_nominalSym](#)

Definition at line 12 of file Classes.c.

Referenced by `is_nominal()`, and `party_init()`.

5.1.3.15 SEXP [PL2_is_ordinalSym](#)

Definition at line 12 of file Classes.c.

Referenced by `is_ordinal()`, and `party_init()`.

5.1.3.16 SEXP [PL2_jobuSym](#)

Definition at line 12 of file Classes.c.

Referenced by `party_init()`.

5.1.3.17 SEXP [PL2_jobvSym](#)

Definition at line 12 of file Classes.c.

Referenced by `party_init()`.

5.1.3.18 SEXP [PL2_levelsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_levels()`, and `party_init()`.

5.1.3.19 SEXP [PL2_linearstatisticSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_LinStatExpCov()`, `C_Node()`, `C_TestStatistic()`, `party_init()`, and `R_splitcategorical()`.

5.1.3.20 SEXP [PL2_linexpcov2sampleSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_Node()`, and `party_init()`.

5.1.3.21 SEXP [PL2_maxdepthSym](#)

Definition at line 12 of file Classes.c.

Referenced by `check_depth()`, and `party_init()`.

5.1.3.22 SEXP [PL2_maxptsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_maxpts()`, and `party_init()`.

5.1.3.23 SEXP [PL2_maxsurrogateSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_maxsurrogate()`, and `party_init()`.

5.1.3.24 SEXP [PL2_methodSym](#)

Definition at line 12 of file Classes.c.

Referenced by `party_init()`.

5.1.3.25 SEXP [PL2_minbucketSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_minbucket()`, and `party_init()`.

5.1.3.26 SEXP [PL2_mincriterionSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_mincriterion()`, and `party_init()`.

5.1.3.27 SEXP [PL2_minprobSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_minprob()`, and `party_init()`.

5.1.3.28 SEXP [PL2_minsplitSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_minsplit()`, and `party_init()`.

5.1.3.29 SEXP [PL2_MPinvSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_MPinv()`, `C_TestStatistic()`, `party_init()`, and `R_MPinv()`.

5.1.3.30 SEXP [PL2_mtrySym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_mtry()`, and `party_init()`.

5.1.3.31 SEXP [PL2_ninputsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_ninputs()`, and `party_init()`.

5.1.3.32 SEXP [PL2_nobsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_nobs()`, and `party_init()`.

5.1.3.33 SEXP [PL2_nresampleSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_nresample()`, and `party_init()`.

5.1.3.34 SEXP [PL2_ntreeSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_ntree()`, and `party_init()`.

5.1.3.35 SEXP [PL2_orderingSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_ordering()`, and `party_init()`.

5.1.3.36 SEXP [PL2_predict_trafoSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_predict_trafo()`, and `party_init()`.

5.1.3.37 SEXP [PL2_pSym](#)

Definition at line 12 of file Classes.c.

Referenced by `CR_svd()`, `party_init()`, and `R_MPinv()`.

5.1.3.38 SEXP [PL2_pvalueSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_pvalue()`, and `party_init()`.

5.1.3.39 SEXP [PL2_randomsplitsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_randomsplits()`, and `party_init()`.

5.1.3.40 SEXP [PL2_rankSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_ConditionalPvalue()`, `C_MPinv()`, `party_init()`, and `R_MPinv()`.

5.1.3.41 SEXP PL2_relepsSym

Definition at line 12 of file Classes.c.

Referenced by `get_releps()`, and `party_init()`.

5.1.3.42 SEXP PL2_replaceSym

Definition at line 12 of file Classes.c.

Referenced by `get_replace()`, and `party_init()`.

5.1.3.43 SEXP PL2_responsesSym

Definition at line 12 of file Classes.c.

Referenced by `C_GlobalTest()`, `C_MonteCarlo()`, `C_Node()`, `C_splitnode()`, `party_init()`, `R_get_response()`, `R_Node()`, `R_set_response()`, and `R_TreeGrow()`.

5.1.3.44 SEXP PL2_savesplitstatsSym

Definition at line 12 of file Classes.c.

Referenced by `get_savesplitstats()`, and `party_init()`.

5.1.3.45 SEXP PL2_scoresSym

Definition at line 12 of file Classes.c.

Referenced by `get_scores()`, and `party_init()`.

5.1.3.46 SEXP PL2_splitctrlSym

Definition at line 12 of file Classes.c.

Referenced by `get_splitctrl()`, and `party_init()`.

5.1.3.47 SEXP PL2_splitstatisticsSym

Definition at line 12 of file Classes.c.

Referenced by `get_splitstatistics()`, and `party_init()`.

5.1.3.48 SEXP PL2_sSym

Definition at line 12 of file Classes.c.

Referenced by `C_MPinv()`, and `party_init()`.

5.1.3.49 SEXP PL2_stumpSym

Definition at line 12 of file Classes.c.

Referenced by `get_stump()`, and `party_init()`.

5.1.3.50 **SEXP [PL2_sumweightsSym](#)**

Definition at line 12 of file `Classes.c`.

Referenced by `C_ExpectCovarInfluence()`, `C_ExpectCovarLinearStatistic()`, `C_GlobalTest()`, `C_MonteCarlo()`, `C_Node()`, `party_init()`, and `R_ExpectCovarInfluence()`.

5.1.3.51 **SEXP [PL2_svdmemSym](#)**

Definition at line 12 of file `Classes.c`.

Referenced by `C_LinStatExpCovMPinv()`, and `party_init()`.

5.1.3.52 **SEXP [PL2_test_trafoSym](#)**

Definition at line 12 of file `Classes.c`.

Referenced by `get_test_trafo()`, and `party_init()`.

5.1.3.53 **SEXP [PL2_teststatSym](#)**

Definition at line 12 of file `Classes.c`.

Referenced by `get_teststat()`, and `party_init()`.

5.1.3.54 **SEXP [PL2_testtypeSym](#)**

Definition at line 12 of file `Classes.c`.

Referenced by `get_testtype()`, and `party_init()`.

5.1.3.55 **SEXP [PL2_tgctrlSym](#)**

Definition at line 12 of file `Classes.c`.

Referenced by `get_tgctrl()`, and `party_init()`.

5.1.3.56 **SEXP [PL2_tolSym](#)**

Definition at line 12 of file `Classes.c`.

Referenced by `get_tol()`, and `party_init()`.

5.1.3.57 **SEXP [PL2_transformationsSym](#)**

Definition at line 12 of file `Classes.c`.

Referenced by `get_transformation()`, `party_init()`, and `R_set_response()`.

5.1.3.58 SEXP [PL2_uSym](#)

Definition at line 12 of file Classes.c.

Referenced by C_MPinv(), CR_svd(), and party_init().

5.1.3.59 SEXP [PL2_varctrlSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_varctrl(), and party_init().

5.1.3.60 SEXP [PL2_variablesSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_variable(), party_init(), R_get_response(), and R_set_response().

5.1.3.61 SEXP [PL2_varmemorySym](#)

Definition at line 12 of file Classes.c.

Referenced by get_varmemory(), and party_init().

5.1.3.62 SEXP [PL2_vSym](#)

Definition at line 12 of file Classes.c.

Referenced by C_MPinv(), CR_svd(), and party_init().

5.1.3.63 SEXP [PL2_weightsSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_weights(), and party_init().

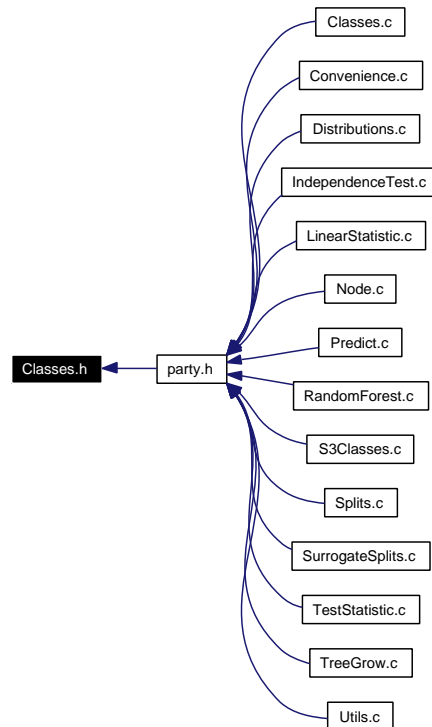
5.1.3.64 SEXP [PL2_whichNASym](#)

Definition at line 12 of file Classes.c.

Referenced by get_missings(), and party_init().

5.2 Classes.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- `int get_dimension` (SEXP object)
- `int get_teststat` (SEXP object)
- `double get_tol` (SEXP object)
- `int get_pvalue` (SEXP object)
- `int get_maxpts` (SEXP object)
- `double get_abseps` (SEXP object)
- `double get_releps` (SEXP object)
- `double get_minsplit` (SEXP object)
- `double get_minprob` (SEXP object)
- `double get_minbucket` (SEXP object)
- `SEXP get_transformation` (SEXP object, int variable)
- `SEXP get_test_trafo` (SEXP object)
- `SEXP get_predict_trafo` (SEXP object)
- `SEXP get_variable` (SEXP object, int variable)
- `int is_nominal` (SEXP object, int variable)
- `int is_ordinal` (SEXP object, int variable)
- `int is_censored` (SEXP object, int variable)
- `int has_missings` (SEXP object, int variable)
- `SEXP get_missings` (SEXP object, int variable)
- `SEXP get_ordering` (SEXP object, int variable)

- SEXP [get_levels](#) (SEXP object, int variable)
- SEXP [get_scores](#) (SEXP object, int variable)
- SEXP [get_whichNA](#) (SEXP object, int variable)
- SEXP [get_varmemory](#) (SEXP object, int variable)
- int [get_nobs](#) (SEXP object)
- int [get_ninputs](#) (SEXP object)
- SEXP [get_weights](#) (SEXP object, int variable)
- int [get_testtype](#) (SEXP object)
- int [get_nresample](#) (SEXP object)
- SEXP [get_varctrl](#) (SEXP object)
- SEXP [get_splitctrl](#) (SEXP object)
- SEXP [get_gtctrl](#) (SEXP object)
- double [get_mincriterion](#) (SEXP object)
- int [get_randomsplits](#) (SEXP object)
- int [get_mtry](#) (SEXP object)
- SEXP [get_dontuse](#) (SEXP object)
- SEXP [get_dontusetmp](#) (SEXP object)
- int [get_stump](#) (SEXP object)
- int [get_maxsurrogate](#) (SEXP object)
- SEXP [get_tgctrl](#) (SEXP object)
- SEXP [get_splitstatistics](#) (SEXP object)
- int [get_savesplitstats](#) (SEXP object)
- int [check_depth](#) (SEXP object, int depth)
- int [get_ntree](#) (SEXP object)
- int [get_replace](#) (SEXP object)
- double [get_fraction](#) (SEXP object)

Variables

- SEXP [PL2_expectationSym](#)
- SEXP [PL2_covarianceSym](#)
- SEXP [PL2_linearstatisticSym](#)
- SEXP [PL2_expcovinfSym](#)
- SEXP [PL2_expcovinfssSym](#)
- SEXP [PL2_sumweightsSym](#)
- SEXP [PL2_dimensionSym](#)
- SEXP [PL2_MPinvSym](#)
- SEXP [PL2_rankSym](#)
- SEXP [PL2_svdmemSym](#)
- SEXP [PL2_methodSym](#)
- SEXP [PL2_jobuSym](#)
- SEXP [PL2_jobvSym](#)
- SEXP [PL2_uSym](#)
- SEXP [PL2_vSym](#)
- SEXP [PL2_sSym](#)
- SEXP [PL2_pSym](#)
- SEXP [PL2_teststatSym](#)
- SEXP [PL2_pvalueSym](#)
- SEXP [PL2_tolSym](#)
- SEXP [PL2_maxptsSym](#)

- [SEXP PL2_absepsSym](#)
- [SEXP PL2_relepsSym](#)
- [SEXP PL2_minsplitSym](#)
- [SEXP PL2_minbucketSym](#)
- [SEXP PL2_minprobSym](#)
- [SEXP PL2_variablesSym](#)
- [SEXP PL2_transformationsSym](#)
- [SEXP PL2_is_nominalSym](#)
- [SEXP PL2_is_ordinalSym](#)
- [SEXP PL2_is_censoredSym](#)
- [SEXP PL2_orderingSym](#)
- [SEXP PL2_levelsSym](#)
- [SEXP PL2_scoresSym](#)
- [SEXP PL2_has_missingsSym](#)
- [SEXP PL2_whichNASym](#)
- [SEXP PL2_test_trafoSym](#)
- [SEXP PL2_predict_trafoSym](#)
- [SEXP PL2_nobsSym](#)
- [SEXP PL2_ninputsSym](#)
- [SEXP PL2_linexpcov2sampleSym](#)
- [SEXP PL2_weightsSym](#)
- [SEXP PL2_varmemorySym](#)
- [SEXP PL2_linexpcov2sampleSym](#)
- [SEXP PL2_weightsSym](#)
- [SEXP PL2_varmemorySym](#)
- [SEXP PL2_responsesSym](#)
- [SEXP PL2_inputsSym](#)
- [SEXP PL2_testtypeSym](#)
- [SEXP PL2_nresampleSym](#)
- [SEXP PL2_varctrlSym](#)
- [SEXP PL2_splitctrlSym](#)
- [SEXP PL2_gtctrlSym](#)
- [SEXP PL2_mincriterionSym](#)
- [SEXP PL2_randomsplitsSym](#)
- [SEXP PL2_mtrySym](#)
- [SEXP PL2_dontuseSym](#)
- [SEXP PL2_dontusetmpSym](#)
- [SEXP PL2_stumpSym](#)
- [SEXP PL2_tgctrlSym](#)
- [SEXP PL2_ntreeSym](#)
- [SEXP PL2_replaceSym](#)
- [SEXP PL2_fractionSym](#)

5.2.1 Function Documentation

5.2.1.1 `int check_depth (SEXP object, int depth)`

Definition at line 346 of file `Classes.c`.

References [PL2_maxdepthSym](#).

Referenced by [C_TreeGrow\(\)](#).

5.2.1.2 double get_abseps (SEXP object)

Definition at line 167 of file Classes.c.

References PL2_absepsSym.

Referenced by C_TeststatPvalue().

5.2.1.3 int get_dimension (SEXP object)

Definition at line 147 of file Classes.c.

References PL2_dimensionSym.

Referenced by C_ConditionalPvalue(), C_Node(), C_TestStatistic(), and R_splitcategorical().

5.2.1.4 SEXP get_dontuse (SEXP object)

Definition at line 334 of file Classes.c.

References PL2_dontuseSym.

5.2.1.5 SEXP get_dontusetmp (SEXP object)

Definition at line 338 of file Classes.c.

References PL2_dontusetmpSym.

5.2.1.6 double get_fraction (SEXP object)

Definition at line 362 of file Classes.c.

References PL2_fractionSym.

5.2.1.7 SEXP get_gtctrl (SEXP object)

Definition at line 310 of file Classes.c.

References PL2_gtctrlSym.

Referenced by C_Node().

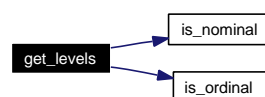
5.2.1.8 SEXP get_levels (SEXP object, int variable)

Definition at line 235 of file Classes.c.

References is_nominal(), is_ordinal(), and PL2_levelsSym.

Referenced by C_Node().

Here is the call graph for this function:



5.2.1.9 int get_maxpts (SEXP object)

Definition at line 163 of file Classes.c.

References PL2_maxptsSym.

Referenced by C_TeststatPvalue().

5.2.1.10 int get_maxsurrogate (SEXP object)

Definition at line 322 of file Classes.c.

References PL2_maxsurrogateSym.

Referenced by C_splitnode(), C_surrogates(), C_TreeGrow(), R_Node(), and R_TreeGrow().

5.2.1.11 double get_minbucket (SEXP object)

Definition at line 183 of file Classes.c.

References PL2_minbucketSym.

5.2.1.12 double get_mincriterion (SEXP object)

Definition at line 318 of file Classes.c.

References PL2_mincriterionSym.

Referenced by C_Node().

5.2.1.13 double get_minprob (SEXP object)

Definition at line 179 of file Classes.c.

References PL2_minprobSym.

5.2.1.14 double get_minsplit (SEXP object)

Definition at line 175 of file Classes.c.

References PL2_minsplitSym.

Referenced by C_Node().

5.2.1.15 SEXP get_missings (SEXP object, int variable)

Definition at line 258 of file Classes.c.

References has_missings(), and PL2_whichNASym.

Referenced by C_get_node(), C_splitnode(), C_splitsurrogate(), and C_surrogates().

Here is the call graph for this function:



5.2.1.16 int get_mtry (SEXP *object*)

Definition at line 330 of file Classes.c.

References PL2_mtrySym.

5.2.1.17 int get_ninputs (SEXP *object*)

Definition at line 286 of file Classes.c.

References PL2_ninputsSym.

Referenced by C_GlobalTest(), C_MonteCarlo(), C_Node(), C_splitnode(), C_surrogates(), R_GlobalTest(), R_MonteCarlo(), R_Node(), and R_TreeGrow().

5.2.1.18 int get_nobs (SEXP *object*)

Definition at line 282 of file Classes.c.

References PL2_nobsSym.

Referenced by C_GlobalTest(), C_MonteCarlo(), C_Node(), C_predict(), C_splitnode(), C_splitsurrogate(), C_surrogates(), C_TreeGrow(), R_Ensemble(), R_get_nodeID(), R_Node(), R_predict(), R_predictRF_weights(), and R_TreeGrow().

5.2.1.19 int get_nresample (SEXP *object*)

Definition at line 298 of file Classes.c.

References PL2_nresampleSym.

Referenced by C_MonteCarlo().

5.2.1.20 int get_ntree (SEXP *object*)

Definition at line 354 of file Classes.c.

References PL2_ntreeSym.

Referenced by R_Ensemble().

5.2.1.21 SEXP get_ordering (SEXP *object*, int *variable*)

Definition at line 224 of file Classes.c.

References is_nominal(), and PL2_orderingSym.

Referenced by C_Node().

Here is the call graph for this function:



5.2.1.22 SEXP get_predict_trafo (SEXP object)

Definition at line 197 of file Classes.c.

References PL2_predict_trafoSym.

Referenced by C_Node(), C_splitnode(), R_modify_response(), R_Node(), R_set_response(), and R_TreeGrow().

5.2.1.23 int get_pvalue (SEXP object)

Definition at line 155 of file Classes.c.

References PL2_pvalueSym.

Referenced by C_TeststatCriterion(), and C_TeststatPvalue().

5.2.1.24 int get_randomsplits (SEXP object)

Definition at line 326 of file Classes.c.

References PL2_randomsplitsSym.

5.2.1.25 double get_releps (SEXP object)

Definition at line 171 of file Classes.c.

References PL2_relepsSym.

Referenced by C_TeststatPvalue().

5.2.1.26 int get_replace (SEXP object)

Definition at line 358 of file Classes.c.

References PL2_replaceSym.

5.2.1.27 int get_savesplitstats (SEXP object)

Definition at line 274 of file Classes.c.

References PL2_savesplitstatsSym.

Referenced by C_Node().

5.2.1.28 SEXP get_scores (SEXP object, int variable)

Definition at line 247 of file Classes.c.

References is_ordinal(), and PL2_scoresSym.

Here is the call graph for this function:



5.2.1.29 SEXP get_splitctrl (SEXP object)

Definition at line 306 of file Classes.c.

References PL2_splitctrlSym.

Referenced by C_Node(), C_splitnode(), C_surrogates(), C_TreeGrow(), R_Node(), and R_TreeGrow().

5.2.1.30 SEXP get_splitstatistics (SEXP object)

Definition at line 278 of file Classes.c.

References PL2_splitstatisticsSym.

Referenced by C_Node().

5.2.1.31 int get_stump (SEXP object)

Definition at line 342 of file Classes.c.

References PL2_stumpSym.

Referenced by C_TreeGrow().

5.2.1.32 SEXP get_test_trafo (SEXP object)

Definition at line 193 of file Classes.c.

References PL2_test_trafoSym.

Referenced by C_GlobalTest(), C_MonteCarlo(), C_Node(), R_modify_response(), and R_set_response().

5.2.1.33 int get_teststat (SEXP object)

Definition at line 151 of file Classes.c.

References PL2_teststatSym.

Referenced by C_IndependenceTest(), and C_TeststatPvalue().

5.2.1.34 int get_testtype (SEXP object)

Definition at line 294 of file Classes.c.

References PL2_testtypeSym.

5.2.1.35 SEXP get_tgctrl (SEXP object)

Definition at line 314 of file Classes.c.

References PL2_tgctrlSym.

Referenced by C_Node(), and C_TreeGrow().

5.2.1.36 double get_tol (SEXP object)

Definition at line 159 of file Classes.c.

References PL2_tolSym.

Referenced by C_IndependenceTest(), C_Node(), C_split(), C_splitcategorical(), C_TeststatPvalue(), and R_splitcategorical().

5.2.1.37 SEXP get_transformation (SEXP object, int variable)

Definition at line 187 of file Classes.c.

References PL2_transformationsSym.

Referenced by C_Node(), and R_modify_response().

5.2.1.38 SEXP get_varctrl (SEXP object)

Definition at line 302 of file Classes.c.

References PL2_varctrlSym.

Referenced by C_Node().

5.2.1.39 SEXP get_variable (SEXP object, int variable)

Definition at line 202 of file Classes.c.

References PL2_variablesSym.

Referenced by C_get_node(), C_Node(), C_splitnode(), C_splitsurrogate(), and R_modify_response().

5.2.1.40 SEXP get_varmemory (SEXP object, int variable)

Definition at line 269 of file Classes.c.

References PL2_varmemorySym.

Referenced by C_Node().

5.2.1.41 SEXP get_weights (SEXP object, int variable)

Definition at line 290 of file Classes.c.

References PL2_weightsSym.

Referenced by C_Node().

5.2.1.42 SEXP get_whichNA (SEXP object, int variable)**5.2.1.43 int has_missings (SEXP object, int variable)**

Definition at line 220 of file Classes.c.

References PL2_has_missingsSym.

Referenced by `C_get_node()`, `C_Node()`, `C_splitnode()`, `C_splitsurrogate()`, `C_surrogates()`, and `get_missings()`.

5.2.1.44 `int is_censored (SEXP object, int variable)`

Definition at line 216 of file `Classes.c`.

References `PL2_is_censoredSym`.

5.2.1.45 `int is_nominal (SEXP object, int variable)`

Definition at line 208 of file `Classes.c`.

References `PL2_is_nominalSym`.

Referenced by `C_Node()`, `get_levels()`, and `get_ordering()`.

5.2.1.46 `int is_ordinal (SEXP object, int variable)`

Definition at line 212 of file `Classes.c`.

References `PL2_is_ordinalSym`.

Referenced by `get_levels()`, and `get_scores()`.

5.2.2 Variable Documentation

5.2.2.1 `SEXP PL2_absepsSym`

Definition at line 12 of file `Classes.c`.

Referenced by `get_abseps()`, and `party_init()`.

5.2.2.2 `SEXP PL2_covarianceSym`

Definition at line 12 of file `Classes.c`.

Referenced by `C_ConditionalPvalue()`, `C_ExpectCovarInfluence()`, `C_ExpectCovarLinearStatistic()`, `C_LinStatExpCovMPinv()`, `C_Node()`, `C_TestStatistic()`, `party_init()`, `R_ExpectCovarInfluence()`, `R_ExpectCovarLinearStatistic()`, and `R_splitcategorical()`.

5.2.2.3 `SEXP PL2_dimensionSym`

Definition at line 12 of file `Classes.c`.

Referenced by `get_dimension()`, and `party_init()`.

5.2.2.4 `SEXP PL2_dontuseSym`

Definition at line 12 of file `Classes.c`.

Referenced by `get_dontuse()`, and `party_init()`.

5.2.2.5 SEXP [PL2_dontusetmpSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_dontusetmp()`, and `party_init()`.

5.2.2.6 SEXP [PL2_expcovinfssSym](#)

Definition at line 12 of file Classes.c.

Referenced by `party_init()`.

5.2.2.7 SEXP [PL2_expcovinfSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_GlobalTest()`, `C_IndependenceTest()`, `C_MonteCarlo()`, `C_Node()`, `party_init()`, and `R_splitcategorical()`.

5.2.2.8 SEXP [PL2_expectationSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_ExpectCovarInfluence()`, `C_ExpectCovarLinearStatistic()`, `C_Node()`, `C_TestStatistic()`, `party_init()`, `R_ExpectCovarInfluence()`, `R_ExpectCovarLinearStatistic()`, and `R_splitcategorical()`.

5.2.2.9 SEXP [PL2_fractionSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_fraction()`, and `party_init()`.

5.2.2.10 SEXP [PL2_gtctrlSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_gtctrl()`, and `party_init()`.

5.2.2.11 SEXP [PL2_has_missingsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `has_missings()`, and `party_init()`.

5.2.2.12 SEXP [PL2_inputsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_GlobalTest()`, `C_MonteCarlo()`, `C_Node()`, `C_splitnode()`, `C_splitsurrogate()`, `C_surrogates()`, and `party_init()`.

5.2.2.13 SEXP PL2_is_censoredSym

Definition at line 12 of file Classes.c.

Referenced by `is_censored()`, and `party_init()`.

5.2.2.14 SEXP PL2_is_nominalSym

Definition at line 12 of file Classes.c.

Referenced by `is_nominal()`, and `party_init()`.

5.2.2.15 SEXP PL2_is_ordinalSym

Definition at line 12 of file Classes.c.

Referenced by `is_ordinal()`, and `party_init()`.

5.2.2.16 SEXP PL2_jobuSym

Definition at line 12 of file Classes.c.

Referenced by `party_init()`.

5.2.2.17 SEXP PL2_jobvSym

Definition at line 12 of file Classes.c.

Referenced by `party_init()`.

5.2.2.18 SEXP PL2_levelsSym

Definition at line 12 of file Classes.c.

Referenced by `get_levels()`, and `party_init()`.

5.2.2.19 SEXP PL2_linearstatisticSym

Definition at line 12 of file Classes.c.

Referenced by `C_LinStatExpCov()`, `C_Node()`, `C_TestStatistic()`, `party_init()`, and `R_splitcategorical()`.

5.2.2.20 SEXP PL2_linexpcov2sampleSym

Definition at line 12 of file Classes.c.

Referenced by `C_Node()`, and `party_init()`.

5.2.2.21 SEXP PL2_linexpcov2sampleSym

Definition at line 12 of file Classes.c.

5.2.2.22 SEXP [PL2_maxptsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_maxpts()`, and `party_init()`.

5.2.2.23 SEXP [PL2_methodSym](#)

Definition at line 12 of file Classes.c.

Referenced by `party_init()`.

5.2.2.24 SEXP [PL2_minbucketSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_minbucket()`, and `party_init()`.

5.2.2.25 SEXP [PL2_mincriterionSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_mincriterion()`, and `party_init()`.

5.2.2.26 SEXP [PL2_minprobSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_minprob()`, and `party_init()`.

5.2.2.27 SEXP [PL2_minsplitSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_minsplit()`, and `party_init()`.

5.2.2.28 SEXP [PL2_MPinvSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_MPinv()`, `C_TestStatistic()`, `party_init()`, and `R_MPinv()`.

5.2.2.29 SEXP [PL2_mtrySym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_mtry()`, and `party_init()`.

5.2.2.30 SEXP [PL2_ninputsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_ninputs()`, and `party_init()`.

5.2.2.31 SEXP [PL2_nobsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_nobs()`, and `party_init()`.

5.2.2.32 SEXP [PL2_nresampleSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_nresample()`, and `party_init()`.

5.2.2.33 SEXP [PL2_ntreeSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_ntree()`, and `party_init()`.

5.2.2.34 SEXP [PL2_orderingSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_ordering()`, and `party_init()`.

5.2.2.35 SEXP [PL2_predict_trafoSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_predict_trafo()`, and `party_init()`.

5.2.2.36 SEXP [PL2_pSym](#)

Definition at line 12 of file Classes.c.

Referenced by `CR_svd()`, `party_init()`, and `R_MPinv()`.

5.2.2.37 SEXP [PL2_pvalueSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_pvalue()`, and `party_init()`.

5.2.2.38 SEXP [PL2_randomsplitsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_randomsplits()`, and `party_init()`.

5.2.2.39 SEXP [PL2_rankSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_ConditionalPvalue()`, `C_MPinv()`, `party_init()`, and `R_MPinv()`.

5.2.2.40 SEXP [PL2_relepsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_releps()`, and `party_init()`.

5.2.2.41 SEXP [PL2_replaceSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_replace()`, and `party_init()`.

5.2.2.42 SEXP [PL2_responsesSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_GlobalTest()`, `C_MonteCarlo()`, `C_Node()`, `C_splitnode()`, `party_init()`, `R_get_response()`, `R_Node()`, `R_set_response()`, and `R_TreeGrow()`.

5.2.2.43 SEXP [PL2_scoresSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_scores()`, and `party_init()`.

5.2.2.44 SEXP [PL2_splitctrlSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_splitctrl()`, and `party_init()`.

5.2.2.45 SEXP [PL2_sSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_MPinv()`, and `party_init()`.

5.2.2.46 SEXP [PL2_stumpSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_stump()`, and `party_init()`.

5.2.2.47 SEXP [PL2_sumweightsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_ExpectCovarInfluence()`, `C_ExpectCovarLinearStatistic()`, `C_GlobalTest()`, `C_MonteCarlo()`, `C_Node()`, `party_init()`, and `R_ExpectCovarInfluence()`.

5.2.2.48 SEXP [PL2_svdmemSym](#)

Definition at line 12 of file Classes.c.

Referenced by C_LinStatExpCovMPinv(), and party_init().

5.2.2.49 SEXP [PL2_test_trafoSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_test_trafo(), and party_init().

5.2.2.50 SEXP [PL2_teststatSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_teststat(), and party_init().

5.2.2.51 SEXP [PL2_testtypeSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_testtype(), and party_init().

5.2.2.52 SEXP [PL2_tgctrlSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_tgctrl(), and party_init().

5.2.2.53 SEXP [PL2_tolSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_tol(), and party_init().

5.2.2.54 SEXP [PL2_transformationsSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_transformation(), party_init(), and R_set_response().

5.2.2.55 SEXP [PL2_uSym](#)

Definition at line 12 of file Classes.c.

Referenced by C_MPinv(), CR_svd(), and party_init().

5.2.2.56 SEXP [PL2_varctrlSym](#)

Definition at line 12 of file Classes.c.

Referenced by get_varctrl(), and party_init().

5.2.2.57 SEXP [PL2_variablesSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_variable()`, `party_init()`, `R_get_response()`, and `R_set_response()`.

5.2.2.58 SEXP [PL2_varmemorySym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_varmemory()`, and `party_init()`.

5.2.2.59 SEXP [PL2_varmemorySym](#)

Definition at line 12 of file Classes.c.

5.2.2.60 SEXP [PL2_vSym](#)

Definition at line 12 of file Classes.c.

Referenced by `C_MPinv()`, `CR_svd()`, and `party_init()`.

5.2.2.61 SEXP [PL2_weightsSym](#)

Definition at line 12 of file Classes.c.

Referenced by `get_weights()`, and `party_init()`.

5.2.2.62 SEXP [PL2_weightsSym](#)

Definition at line 12 of file Classes.c.

5.2.2.63 SEXP [PL2_whichNASym](#)

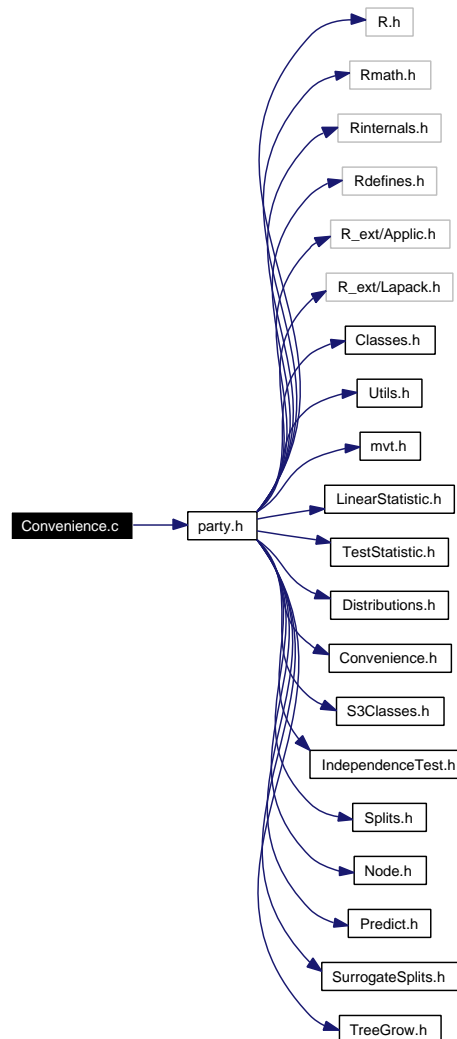
Definition at line 12 of file Classes.c.

Referenced by `get_missings()`, and `party_init()`.

5.3 Convenience.c File Reference

```
#include "party.h"
```

Include dependency graph for Convenience.c:



Functions

- void [C_LinStatExpCov](#) (const double *x, const int p, const double *y, const int q, const double *weights, const int n, const int cexpcovinf, SEXP expcovinf, SEXP ans)
- void [C_LinStatExpCovMPinv](#) (SEXP linexpcov, double tol)
- double [C_TestStatistic](#) (const SEXP linexpcov, const int type, const double tol)
- double [C_ConditionalPvalue](#) (const double tstat, SEXP linexpcov, const int type, double tol, int *maxpts, double *releps, double *abseps)
- SEXP [R_get_response](#) (SEXP learnsample)
- void [R_set_response](#) (SEXP learnsample, SEXP y)

5.3.1 Detailed Description

Some convenience functions

Author

hothorn

Date

2007-02-02 11:22:45 +0100 (Fri, 02 Feb 2007)

Definition in file [Convenience.c](#).

5.3.2 Function Documentation

5.3.2.1 `double C_ConditionalPvalue (const double tstat, SEXP linexpcov, const int type, double tol, int * maxpts, double * releps, double * abseps)`

Compute asymptotic conditional P-value

Parameters:

tstat test statistic

linexpcov an object of class ‘LinStatExpectCovar’

type integer, 1 (maxabs) or 2 (quadform)

tol tolerance

maxpts argument to C_maxabsConditionalPvalue

releps argument to C_maxabsConditionalPvalue

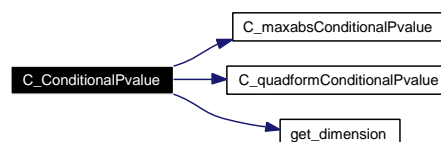
abseps argument to C_maxabsConditionalPvalue

Definition at line 99 of file Convenience.c.

References C_maxabsConditionalPvalue(), C_quadformConditionalPvalue(), get_dimension(), MAXABS, PL2_covarianceSym, PL2_rankSym, and QUADFORM.

Referenced by C_TeststatPvalue().

Here is the call graph for this function:



5.3.2.2 `void C_LinStatExpCov (const double * x, const int p, const double * y, const int q, const double * weights, const int n, const int cexpcovinf, SEXP expcovinf, SEXP ans)`

Linear statistic of *x*, *y*, and *weights* and its conditional expectation and covariance

Parameters:

x values of the transformation

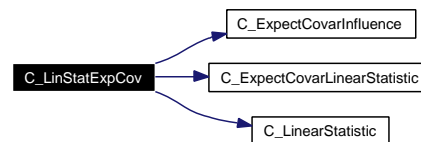
p dimension of the transformation
 y values of the influence function
 q dimension of the influence function
weights case weights
 n number of observations
cexpcovinf logical: recompute exp and cov of the influence fct
expcovinf an object of class 'ExpectCovarInfluence'
ans return value; an object of class 'LinStatExpectCovar'

Definition at line 26 of file Convenience.c.

References C_ExpectCovarInfluence(), C_ExpectCovarLinearStatistic(), C_LinearStatistic(), and PL2_linearstatisticSym.

Referenced by C_IndependenceTest(), and R_splitcategorical().

Here is the call graph for this function:



5.3.2.3 void C_LinStatExpCovMPinv (SEXP *linexpcov*, double *tol*)

Moore-Penrose inverse of the covariance matrix

Parameters:

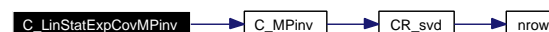
linexpcov an object of class 'LinStatExpectCovarMPinv'
tol tolerance

Definition at line 46 of file Convenience.c.

References C_MPinv(), PL2_covarianceSym, and PL2_svdmemSym.

Referenced by C_IndependenceTest().

Here is the call graph for this function:



5.3.2.4 double C_TestStatistic (const SEXP *linexpcov*, const int *type*, const double *tol*)

Compute test statistic

Parameters:

linexpcov an object of class 'LinStatExpectCovar'
type integer, 1 (maxabs) or 2 (quadform)

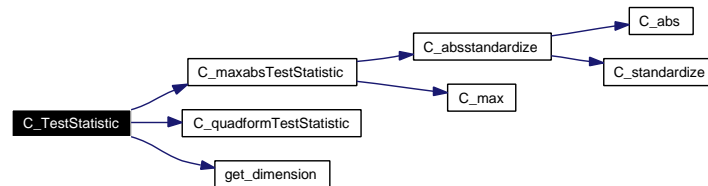
tol tolerance

Definition at line 59 of file Convenience.c.

References C_maxabsTestStatistic(), C_quadformTestStatistic(), get_dimension(), PL2_covarianceSym, PL2_expectationSym, PL2_linearstatisticSym, and PL2_MPinvSym.

Referenced by C_TeststatPvalue().

Here is the call graph for this function:



5.3.2.5 SEXP R_get_response (SEXP *learnsample*)

extract the (first) response variable from a learning sample

Parameters:

learnsample an object of class ‘LearningSample’

Definition at line 131 of file Convenience.c.

References PL2_responsesSym, and PL2_variablesSym.

Referenced by R_set_response().

5.3.2.6 void R_set_response (SEXP *learnsample*, SEXP *y*)

change the values of the response variable in a learning sample

Parameters:

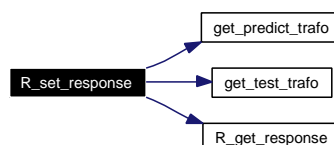
learnsample an object of class ‘LearningSample’

y a REAL with new values

Definition at line 143 of file Convenience.c.

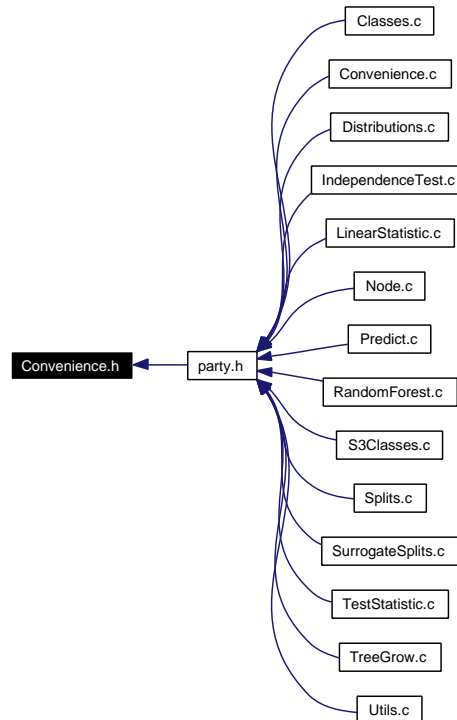
References get_predict_trafo(), get_test_trafo(), PL2_responsesSym, PL2_transformationsSym, PL2_variablesSym, and R_get_response().

Here is the call graph for this function:



5.4 Convenience.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_LinStatExpCov](#) (const double *x, const int p, const double *y, const int q, const double *weights, const int n, const int cexpcovinf, SEXP expcovinf, SEXP ans)
- void [C_LinStatExpCovMPinv](#) (SEXP linexpcov, double tol)
- void [C_MLinearStatistic](#) (SEXP linexpcov, SEXP ScoreMatrix, SEXP ans)
- double [C_TestStatistic](#) (const SEXP linexpcov, const int type, const double tol)
- double [C_ConditionalPvalue](#) (const double tstat, SEXP linexpcov, const int type, double tol, int *maxpts, double *releps, double *abseps)

5.4.1 Function Documentation

5.4.1.1 double C_ConditionalPvalue (const double *tstat*, SEXP *linexpcov*, const int *type*, double *tol*, int * *maxpts*, double * *releps*, double * *abseps*)

Compute asymptotic conditional P-value

Parameters:

- tstat* test statistic
- linexpcov* an object of class ‘LinStatExpectCovar’
- type* integer, 1 (maxabs) or 2 (quadform)
- tol* tolerance

maxpts argument to C_maxabsConditionalPvalue

releps argument to C_maxabsConditionalPvalue

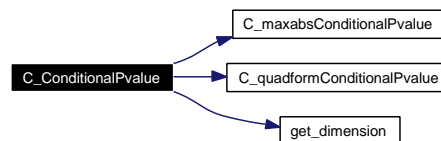
abseps argument to C_maxabsConditionalPvalue

Definition at line 99 of file Convenience.c.

References C_maxabsConditionalPvalue(), C_quadformConditionalPvalue(), get_dimension(), MAXABS, PL2_covarianceSym, PL2_rankSym, and QUADFORM.

Referenced by C_TeststatPvalue().

Here is the call graph for this function:



5.4.1.2 void C_LinStatExpCov (const double * x, const int p, const double * y, const int q, const double * weights, const int n, const int cexpcovinf, SEXP expcovinf, SEXP ans)

Linear statistic of x, y, and weights and its conditional expectation and covariance

Parameters:

x values of the transformation

p dimension of the transformation

y values of the influence function

q dimension of the influence function

weights case weights

n number of observations

cexpcovinf logical: recompute exp and cov of the influence fct

expcovinf an object of class 'ExpectCovarInfluence'

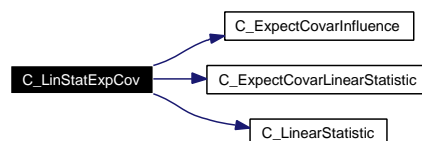
ans return value; an object of class 'LinStatExpectCovar'

Definition at line 26 of file Convenience.c.

References C_ExpectCovarInfluence(), C_ExpectCovarLinearStatistic(), C_LinearStatistic(), and PL2_linearstatisticSym.

Referenced by C_IndependenceTest(), and R_splitcategorical().

Here is the call graph for this function:



5.4.1.3 void C_LinStatExpCovMPinv (SEXP *linexpcov*, double *tol*)

Moore-Penrose inverse of the covariance matrix

Parameters:

linexpcov an object of class ‘LinStatExpectCovarMPinv’

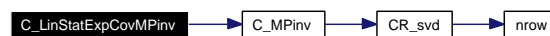
tol tolerance

Definition at line 46 of file Convenience.c.

References C_MPinv(), PL2_covarianceSym, and PL2_svdmemSym.

Referenced by C_IndependenceTest().

Here is the call graph for this function:



5.4.1.4 void C_MLinearStatistic (SEXP *linexpcov*, SEXP *ScoreMatrix*, SEXP *ans*)

5.4.1.5 double C_TestStatistic (const SEXP *linexpcov*, const int *type*, const double *tol*)

Compute test statistic

Parameters:

linexpcov an object of class ‘LinStatExpectCovar’

type integer, 1 (maxabs) or 2 (quadform)

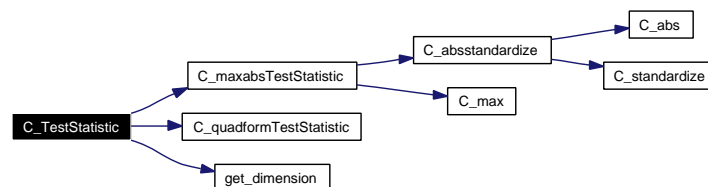
tol tolerance

Definition at line 59 of file Convenience.c.

References C_maxabsTestStatistic(), C_quadformTestStatistic(), get_dimension(), PL2_covarianceSym, PL2_expectationSym, PL2_linearstatisticSym, and PL2_MPinvSym.

Referenced by C_TeststatPvalue().

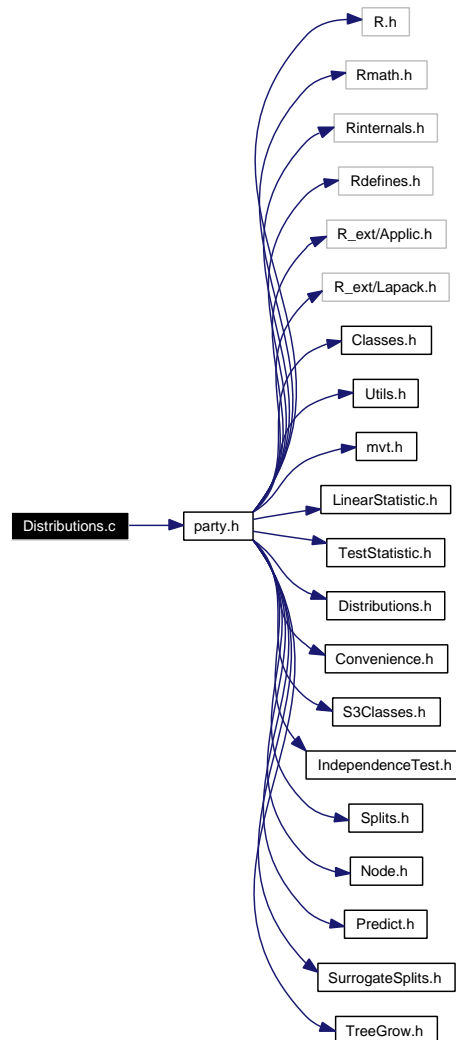
Here is the call graph for this function:



5.5 Distributions.c File Reference

```
#include "party.h"
```

Include dependency graph for Distributions.c:



Functions

- double [C_quadformConditionalPvalue](#) (const double tstat, const double df)
- SEXP [R_quadformConditionalPvalue](#) (SEXP tstat, SEXP df)
- double [C_maxabsConditionalPvalue](#) (const double tstat, const double *Sigma, const int pq, int *maxpts, double *releps, double *abseps, double *tol)
- SEXP [R_maxabsConditionalPvalue](#) (SEXP tstat, SEXP Sigma, SEXP maxpts, SEXP releps, SEXP abseps, SEXP tol)
- void [C_MonteCarlo](#) (double *criterion, SEXP learnsample, SEXP weights, SEXP fitmem, SEXP varctrl, SEXP gctrl, double *ans_pvalues)
- SEXP [R_MonteCarlo](#) (SEXP criterion, SEXP learnsample, SEXP weights, SEXP fitmem, SEXP varctrl, SEXP gctrl)

5.5.1 Detailed Description

Conditional Distributions

Author

hothorn

Date

2007-07-12 18:49:01 +0200 (Thu, 12 Jul 2007)

Definition in file [Distributions.c](#).

5.5.2 Function Documentation

5.5.2.1 `double C_maxabsConditionalPvalue (const double tstat, const double * Sigma, const int pq, int * maxpts, double * releps, double * abseps, double * tol)`

Conditional asymptotic P-value of a maxabs-type test statistic

Basically the functionality from package ‘mvtnorm’

Parameters:

tstat test statistic

Sigma covariance matrix

pq nrow(Sigma)

maxpts number of Monte-Carlo steps

releps relative error

abseps absolute error

tol tolerance

Definition at line 52 of file Distributions.c.

Referenced by C_ConditionalPvalue(), and R_maxabsConditionalPvalue().

5.5.2.2 `void C_MonteCarlo (double * criterion, SEXP learnsample, SEXP weights, SEXP fitmem, SEXP varctrl, SEXP gtctrl, double * ans_pvalues)`

Monte-Carlo approximation to the conditional pvalues

Parameters:

criterion vector of node criteria for each input

learnsample an object of class ‘LearningSample’

weights case weights

fitmem an object of class ‘TreeFitMemory’

varctrl an object of class ‘VariableControl’

gtctrl an object of class ‘GlobalTestControl’

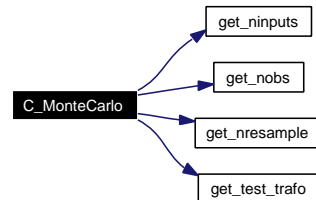
ans_pvalues return values; vector of adjusted pvalues

Definition at line 169 of file Distributions.c.

References `get_ninputs()`, `get_nobs()`, `get_nresample()`, `get_test_trafo()`, `PL2_expcovinfSym`, `PL2_inputsSym`, `PL2_responsesSym`, and `PL2_sumweightsSym`.

Referenced by `R_MonteCarlo()`.

Here is the call graph for this function:



5.5.2.3 double C_quadformConditionalPvalue (const double *tstat*, const double *df*)

Conditional asymptotic P-value of a quadratic form

Parameters:

tstat test statistic

df degree of freedom

Definition at line 18 of file Distributions.c.

Referenced by `C_ConditionalPvalue()`, and `R_quadformConditionalPvalue()`.

5.5.2.4 SEXP R_maxabsConditionalPvalue (SEXP *tstat*, SEXP *Sigma*, SEXP *maxpts*, SEXP *releps*, SEXP *abseps*, SEXP *tol*)

R-interface to `C_maxabsConditionalPvalue`

Parameters:

tstat test statistic

Sigma covariance matrix

maxpts number of Monte-Carlo steps

releps relative error

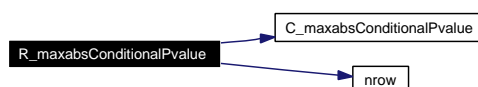
abseps absolute error

tol tolerance

Definition at line 142 of file Distributions.c.

References `C_maxabsConditionalPvalue()`, and `nrow()`.

Here is the call graph for this function:



5.5.2.5 SEXP R_MonteCarlo (SEXP *criterion*, SEXP *learnsample*, SEXP *weights*, SEXP *fitmem*, SEXP *varctrl*, SEXP *gtctrl*)

R-interface to C_MonteCarlo

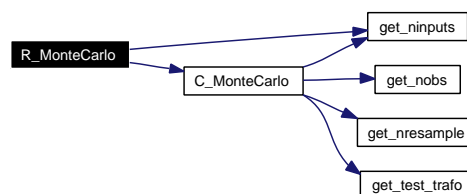
Parameters:

- criterion* vector of node criteria for each input
- learnsample* an object of class 'LearningSample'
- weights* case weights
- fitmem* an object of class 'TreeFitMemory'
- varctrl* an object of class 'VariableControl'
- gtctrl* an object of class 'GlobalTestControl'

Definition at line 278 of file Distributions.c.

References C_MonteCarlo(), and get_ninputs().

Here is the call graph for this function:



5.5.2.6 SEXP R_quadformConditionalPvalue (SEXP *tstat*, SEXP *df*)

R-interface to C_quadformConditionalPvalue

Parameters:

- tstat* test statistic
- df* degree of freedom

Definition at line 29 of file Distributions.c.

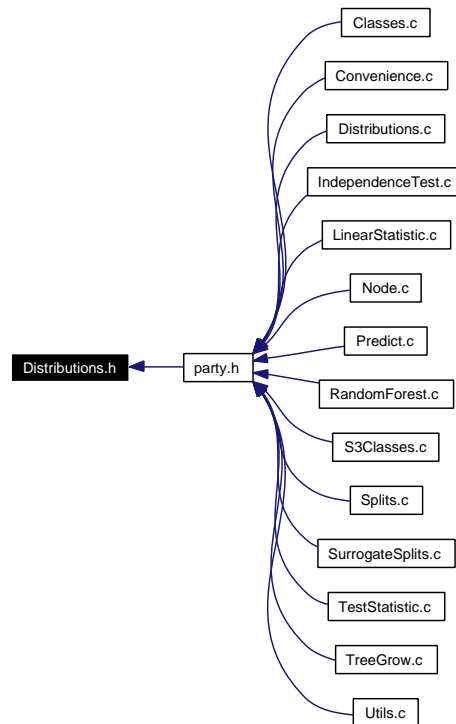
References C_quadformConditionalPvalue().

Here is the call graph for this function:



5.6 Distributions.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- double [C_quadformConditionalPvalue](#) (const double *tstat*, const double *df*)
- double [C_maxabsConditionalPvalue](#) (const double *tstat*, const double **Sigma*, const int *pq*, int **maxpts*, double **releps*, double **abseps*, double **tol*)
- void [C_MonteCarlo](#) (double **pvalues*, SEXP *learnsample*, SEXP *weights*, SEXP *fitmem*, SEXP *varctrl*, SEXP *gtctrl*, double **ans*)

5.6.1 Function Documentation

5.6.1.1 double C_maxabsConditionalPvalue (const double *tstat*, const double * *Sigma*, const int *pq*, int * *maxpts*, double * *releps*, double * *abseps*, double * *tol*)

Conditional asymptotic P-value of a maxabs-type test statistic

Basically the functionality from package ‘mvtnorm’

Parameters:

tstat test statistic

Sigma covariance matrix

pq nrow(*Sigma*)

maxpts number of Monte-Carlo steps

releps relative error
abseps absolute error
tol tolerance

Definition at line 52 of file Distributions.c.

Referenced by C_ConditionalPvalue(), and R_maxabsConditionalPvalue().

5.6.1.2 void C_MonteCarlo (double * *criterion*, SEXP *learnsample*, SEXP *weights*, SEXP *fitmem*, SEXP *varctrl*, SEXP *gtctrl*, double * *ans_pvalues*)

Monte-Carlo approximation to the conditional pvalues

Parameters:

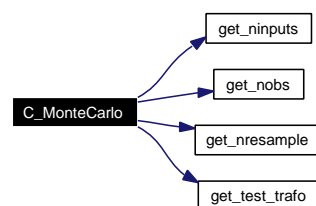
criterion vector of node criteria for each input
learnsample an object of class ‘LearningSample’
weights case weights
fitmem an object of class ‘TreeFitMemory’
varctrl an object of class ‘VariableControl’
gtctrl an object of class ‘GlobalTestControl’
ans_pvalues return values; vector of adjusted pvalues

Definition at line 169 of file Distributions.c.

References get_ninputs(), get_nobs(), get_nresample(), get_test_trafo(), PL2_expcovinfSym, PL2_inputsSym, PL2_responsesSym, and PL2_sumweightsSym.

Referenced by R_MonteCarlo().

Here is the call graph for this function:



5.6.1.3 double C_quadformConditionalPvalue (const double *tstat*, const double *df*)

Conditional asymptotic P-value of a quadratic form

Parameters:

tstat test statistic
df degree of freedom

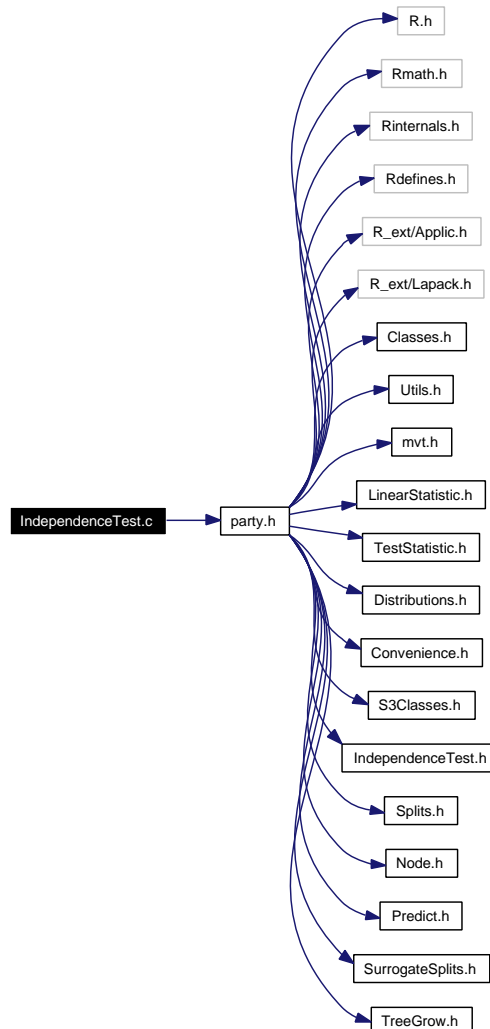
Definition at line 18 of file Distributions.c.

Referenced by C_ConditionalPvalue(), and R_quadformConditionalPvalue().

5.7 IndependenceTest.c File Reference

```
#include "party.h"
```

Include dependency graph for IndependenceTest.c:



Functions

- void [C_TeststatPvalue](#) (const SEXP linexpcov, const SEXP varctrl, double *ans_teststat, double *ans_pvalue)
- void [C_TeststatCriterion](#) (const SEXP linexpcov, const SEXP varctrl, double *ans_teststat, double *ans_criterion)
- void [C_IndependenceTest](#) (const SEXP x, const SEXP y, const SEXP weights, SEXP linexpcov, SEXP varctrl, SEXP ans)
- SEXP [R_IndependenceTest](#) (SEXP x, SEXP y, SEXP weights, SEXP linexpcov, SEXP varctrl)
- void [C_GlobalTest](#) (const SEXP learnsample, const SEXP weights, SEXP fitmem, const SEXP varctrl, const SEXP gctrl, const double minsplit, double *ans_teststat, double *ans_criterion)
- SEXP [R_GlobalTest](#) (SEXP learnsample, SEXP weights, SEXP fitmem, SEXP varctrl, SEXP gctrl)

5.7.1 Detailed Description

Functions for variable selection in each node of a tree

Author

hothorn

Date

2007-02-02 11:22:45 +0100 (Fri, 02 Feb 2007)

Definition in file [IndependenceTest.c](#).

5.7.2 Function Documentation

5.7.2.1 void C_GlobalTest (const SEXP *learnsample*, const SEXP *weights*, SEXP *fitmem*, const SEXP *varctrl*, const SEXP *gtctrl*, const double *minsplit*, double * *ans_teststat*, double * *ans_criterion*)

Perform a global test on independence of a response and multiple inputs

Parameters:

learnsample an object of class ‘LearningSample’

weights case weights

fitmem an object of class ‘TreeFitMemory’

varctrl an object of class ‘VariableControl’

gtctrl an object of class ‘GlobalTestControl’

minsplit minimum sum of weights to proceed

ans_teststat return value; vector of test statistics

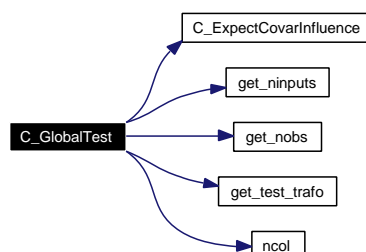
ans_criterion return value; vector of node criteria (adjusted) pvalues or raw test statistics

Definition at line 129 of file IndependenceTest.c.

References C_ExpectCovarInfluence(), get_ninputs(), get_nobs(), get_test_trafo(), ncol(), PL2_expcovinfSym, PL2_inputsSym, PL2_responsesSym, and PL2_sumweightsSym.

Referenced by C_Node(), and R_GlobalTest().

Here is the call graph for this function:



5.7.2.2 void C_IndependenceTest (const SEXP x, const SEXP y, const SEXP weights, SEXP linexpcov, SEXP varctrl, SEXP ans)

Test of independence between x and y

Parameters:

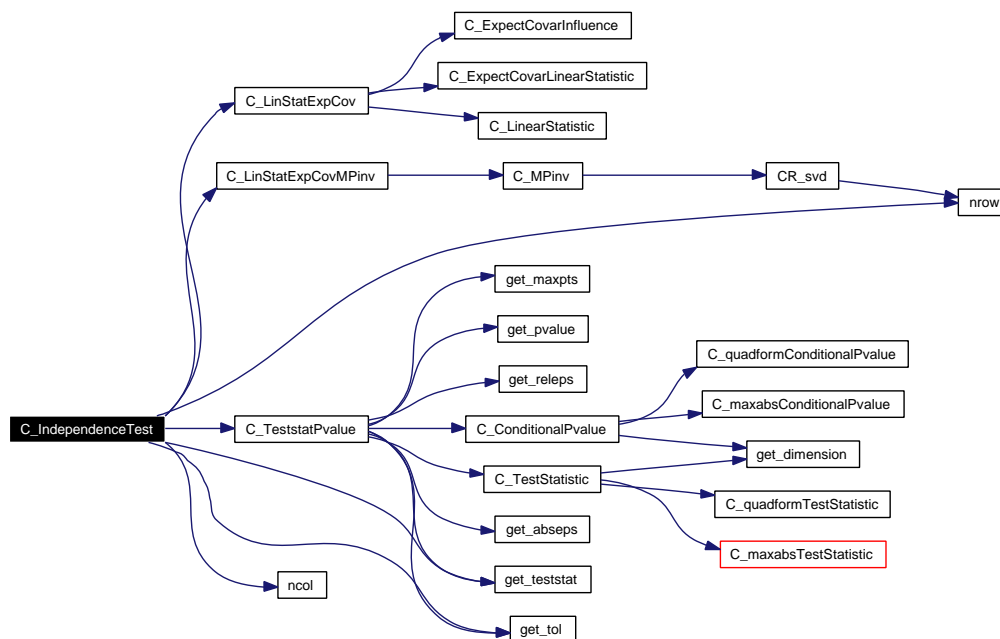
- x** values of the transformation
- y** values of the influence function
- weights** case weights
- linexpcov** an object of class ‘VariableControl’ for T
- varctrl** an object of class ‘VariableControl’
- ans;** return value, a double vector (teststat, pvalue)

Definition at line 78 of file IndependenceTest.c.

References C_LinStatExpCov(), C_LinStatExpCovMPinv(), C_TeststatPvalue(), get_teststat(), get_tol(), ncol(), nrow(), and PL2_expcovinfSym.

Referenced by R_IndependenceTest().

Here is the call graph for this function:



5.7.2.3 void C_TeststatCriterion (const SEXP linexpcov, const SEXP varctrl, double * ans_teststat, double * ans_criterion)

Computes the test statistic and the node criterion

Parameters:

- linexpcov** an object of class ‘LinStatExpectCovar’
- varctrl** an object of class ‘VariableControl’

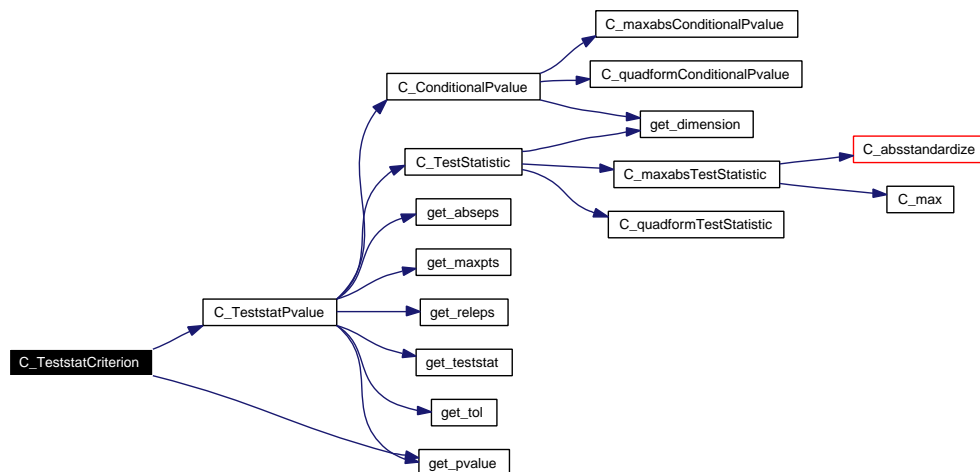
ans_teststat; return value, the test statistic

ans_criterion; return value, the p-value

Definition at line 53 of file IndependenceTest.c.

References C_TeststatPvalue(), and get_pvalue().

Here is the call graph for this function:



5.7.2.4 void C_TeststatPvalue (const SEXP linexpcov, const SEXP varctrl, double * ans_teststat, double * ans_pvalue)

Computes the test statistic and, if requested, the corresponding P-value for a linear statistic

Parameters:

linexpcov an object of class ‘LinStatExpectCovar’

varctrl an object of class ‘VariableControl’

ans_teststat; return value, the test statistic

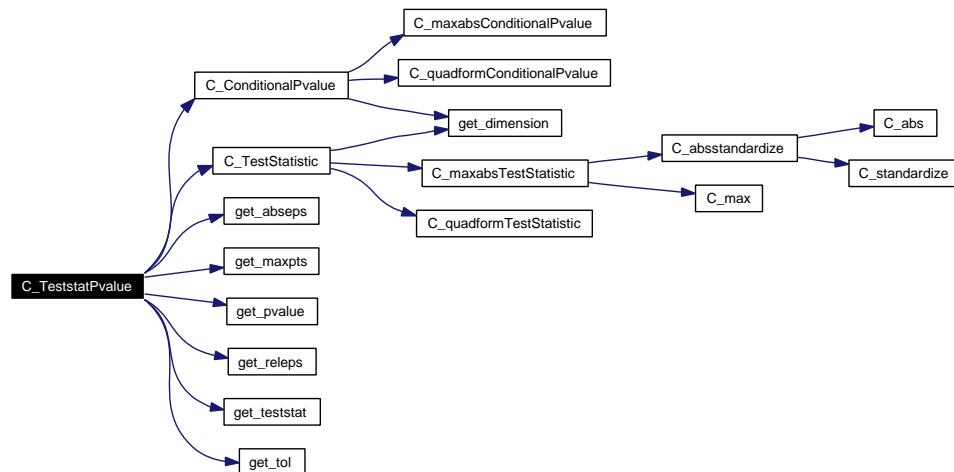
ans_pvalue; return value, the p-value

Definition at line 21 of file IndependenceTest.c.

References C_ConditionalPvalue(), C_TestStatistic(), get_abseps(), get_maxpts(), get_pvalue(), get_releps(), get_teststat(), and get_tol().

Referenced by C_IndependenceTest(), and C_TeststatCriterion().

Here is the call graph for this function:



5.7.2.5 SEXP R_GlobalTest (SEXP *learnsample*, SEXP *weights*, SEXP *fitmem*, SEXP *varctrl*, SEXP *gtctrl*)

R-interface to C_GlobalTest

Parameters:

learnsample an object of class ‘LearningSample’

weights case weights

fitmem an object of class ‘TreeFitMemory’

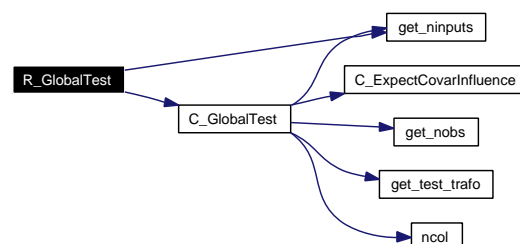
varctrl an object of class ‘VariableControl’

gtctrl an object of class ‘GlobalTestControl’

Definition at line 277 of file IndependenceTest.c.

References C_GlobalTest(), and get_ninputs().

Here is the call graph for this function:



5.7.2.6 SEXP R_IndependenceTest (SEXP *x*, SEXP *y*, SEXP *weights*, SEXP *linexpcov*, SEXP *varctrl*)

R-interface to C_IndependenceTest

Parameters:

x values of the transformation

y values of the influence function

weights case weights

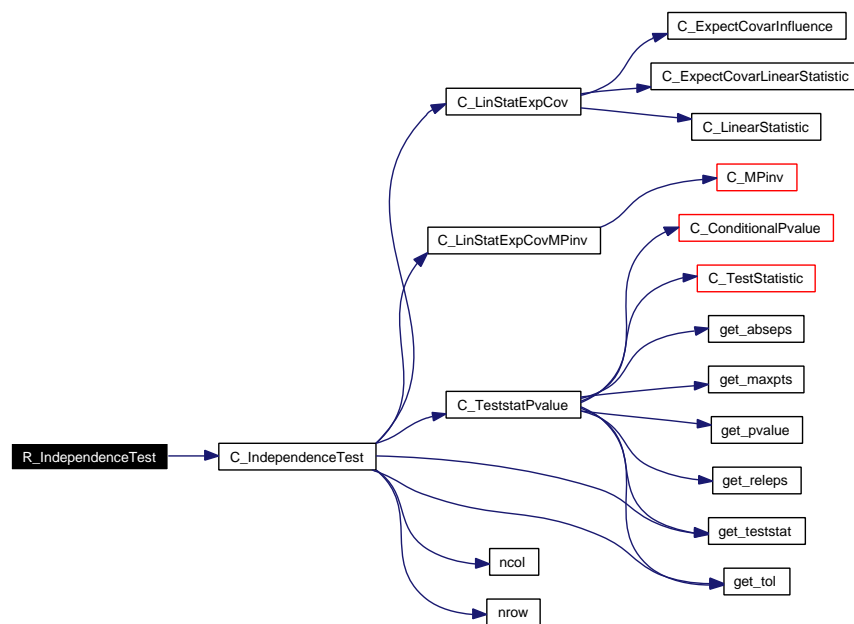
linexpcov an object of class ‘VariableControl’ for T

varctrl an object of class ‘VariableControl’

Definition at line 105 of file IndependenceTest.c.

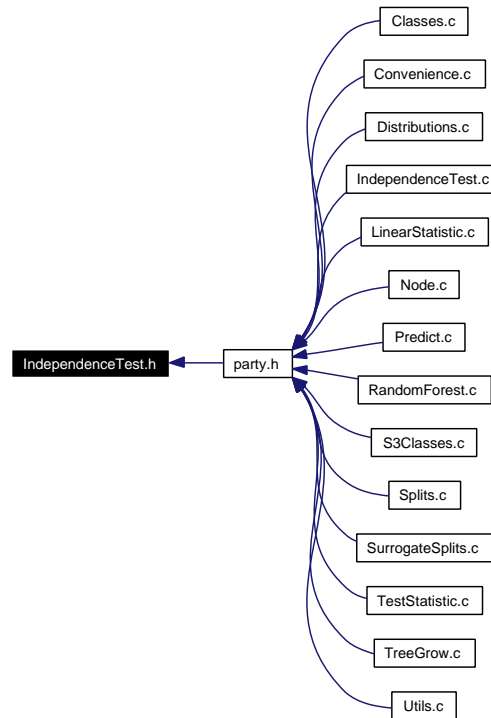
References C_IndependenceTest().

Here is the call graph for this function:



5.8 IndependenceTest.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_GlobalTest](#) (SEXP *learnsample*, SEXP *weights*, SEXP *fitmem*, SEXP *varctrl*, SEXP *gtestctrl*, double *minsplit*, double **teststat*, double **criterion*)
- void [C_TeststatPvalue](#) (const SEXP *linexpcov*, const SEXP *varctrl*, double **ans_teststat*, double **ans_pvalue*)
- void [C_TeststatCriterion](#) (const SEXP *linexpcov*, const SEXP *varctrl*, double **ans_teststat*, double **ans_criterion*)

5.8.1 Function Documentation

5.8.1.1 void [C_GlobalTest](#) (const SEXP *learnsample*, const SEXP *weights*, SEXP *fitmem*, const SEXP *varctrl*, const SEXP *gtctrl*, const double *minsplit*, double **ans_teststat*, double **ans_criterion*)

Perform a global test on independence of a response and multiple inputs

Parameters:

learnsample an object of class ‘LearningSample’
weights case weights
fitmem an object of class ‘TreeFitMemory’
varctrl an object of class ‘VariableControl’

gtctrl an object of class ‘GlobalTestControl’

minsplit minimum sum of weights to proceed

ans_teststat return value; vector of test statistics

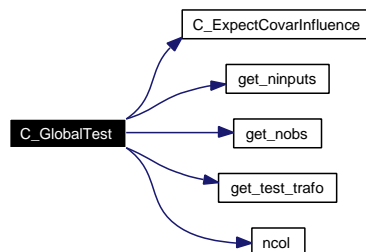
ans_criterion return value; vector of node criteria (adjusted) pvalues or raw test statistics

Definition at line 129 of file IndependenceTest.c.

References C_ExpectCovarInfluence(), get_ninputs(), get_nobs(), get_test_trafo(), ncol(), PL2_expcovinfSym, PL2_inputsSym, PL2_responsesSym, and PL2_sumweightsSym.

Referenced by C_Node(), and R_GlobalTest().

Here is the call graph for this function:



5.8.1.2 void C_TeststatCriterion (const SEXP *linexpcov*, const SEXP *varctrl*, double * *ans_teststat*, double * *ans_criterion*)

Computes the test statistic and the node criterion

Parameters:

linexpcov an object of class ‘LinStatExpectCovar’

varctrl an object of class ‘VariableControl’

ans_teststat; return value, the test statistic

ans_criterion; return value, the p-value

Definition at line 53 of file IndependenceTest.c.

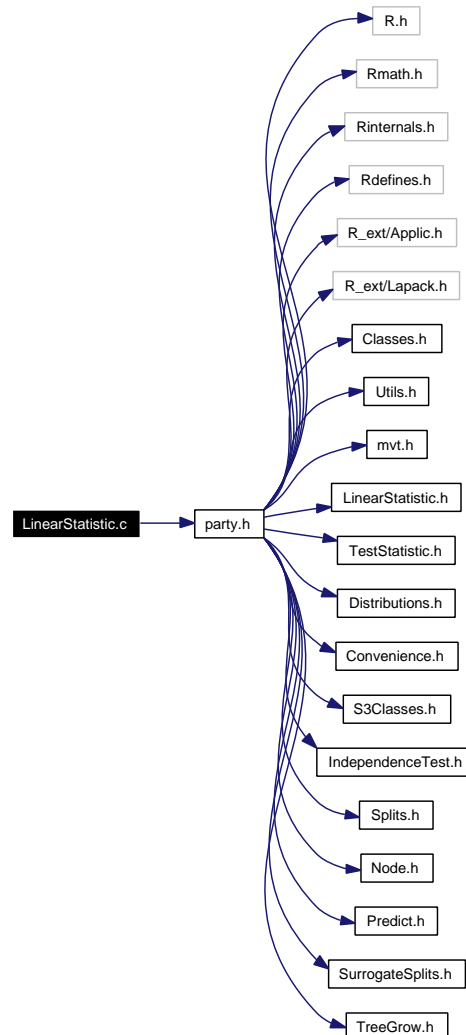
References C_TeststatPvalue(), and get_pvalue().

Here is the call graph for this function:

5.9 LinearStatistic.c File Reference

```
#include "party.h"
```

Include dependency graph for LinearStatistic.c:



Functions

- void [C_LinearStatistic](#) (const double *x, const int p, const double *y, const int q, const double *weights, const int n, double *ans)
- SEXP [R_LinearStatistic](#) (SEXP x, SEXP y, SEXP weights)
- void [C_ExpectCovarInfluence](#) (const double *y, const int q, const double *weights, const int n, SEXP ans)
- SEXP [R_ExpectCovarInfluence](#) (SEXP y, SEXP weights)
- void [C_ExpectCovarLinearStatistic](#) (const double *x, const int p, const double *y, const int q, const double *weights, const int n, const SEXP expcovinf, SEXP ans)
- SEXP [R_ExpectCovarLinearStatistic](#) (SEXP x, SEXP y, SEXP weights, SEXP expcovinf)
- void [C_PermutedLinearStatistic](#) (const double *x, const int p, const double *y, const int q, const int n, const int nperm, const int *indx, const int *perm, double *ans)

- SEXP [R_PermutedLinearStatistic](#) (SEXP *x*, SEXP *y*, SEXP *indx*, SEXP *perm*)

5.9.1 Detailed Description

Linear statistics for conditional inference based on Strasser & Weber (1999)

Author

hothorn

Date

2006-08-25 10:53:10 +0200 (Fri, 25 Aug 2006)

Definition in file [LinearStatistic.c](#).

5.9.2 Function Documentation

5.9.2.1 void C_ExpectCovarInfluence (const double * *y*, const int *q*, const double * *weights*, const int *n*, SEXP *ans*)

Conditional expectation and covariance of the influence function

Parameters:

- y* values of the influence function
- q* dimension of the influence function
- weights* case weights
- n* number of observations
- ans* return value; an object of class 'ExpectCovarInfluence'

Definition at line 101 of file LinearStatistic.c.

References PL2_covarianceSym, PL2_expectationSym, and PL2_sumweightsSym.

Referenced by C_GlobalTest(), C_LinStatExpCov(), and R_ExpectCovarInfluence().

5.9.2.2 void C_ExpectCovarLinearStatistic (const double * *x*, const int *p*, const double * *y*, const int *q*, const double * *weights*, const int *n*, const SEXP *expcovinf*, SEXP *ans*)

Conditional expectation and covariance of the a linear statistic

Parameters:

- x* values of the transformation
- p* dimension of the transformation
- y* values of the influence function
- q* dimension of the influence function
- weights* case weights
- n* number of observations
- expcovinf* an object of class 'ExpectCovarInfluence'
- ans* return value; an object of class 'ExpectCovar'

Definition at line 213 of file LinearStatistic.c.

References PL2_covarianceSym, PL2_expectationSym, and PL2_sumweightsSym.

Referenced by C_LinStatExpCov(), and R_ExpectCovarLinearStatistic().

5.9.2.3 void C_LinearStatistic (const double * *x*, const int *p*, const double * *y*, const int *q*, const double * *weights*, const int *n*, double * *ans*)

Computes the linear statistic, formula (1) in the paper

Parameters:

x values of the transformation
p dimension of the transformation
y values of the influence function
q dimension of the influence function
weights case weights
n number of observations
ans return value; a pointer to a REALSXP-vector of length pq

Definition at line 23 of file LinearStatistic.c.

Referenced by C_LinStatExpCov(), and R_LinearStatistic().

5.9.2.4 void C_PermutedLinearStatistic (const double * *x*, const int *p*, const double * *y*, const int *q*, const int *n*, const int *nperm*, const int * *indx*, const int * *perm*, double * *ans*)

Linear Statistic with permuted indices

Parameters:

x values of the transformation
p dimension of the transformation
y values of the influence function
q dimension of the influence function
n number of observations
nperm number of permutations
indx indices for the x-part
perm (permuted) indices for the y-part
ans return value; a pointer to a REALSXP-vector of length pq

Definition at line 351 of file LinearStatistic.c.

5.9.2.5 SEXP R_ExpectCovarInfluence (SEXP *y*, SEXP *weights*)

R-interface to C_ExpectCovarInfluence

Parameters:

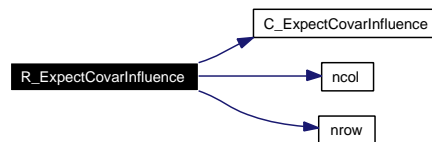
y values of the influence function

weights case weights

Definition at line 171 of file LinearStatistic.c.

References C_ExpectCovarInfluence(), ncol(), nrow(), PL2_covarianceSym, PL2_expectationSym, and PL2_sumweightsSym.

Here is the call graph for this function:



5.9.2.6 SEXP R_ExpectCovarLinearStatistic (SEXP *x*, SEXP *y*, SEXP *weights*, SEXP *expcovinf*)

R-interface to C_ExpectCovarLinearStatistic

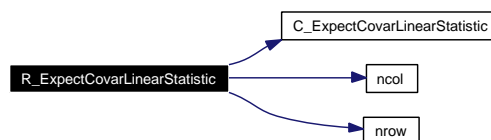
Parameters:

- x* values of the transformation
- y* values of the influence function
- weights* case weights
- expcovinf* an object of class 'ExpectCovarInfluence'

Definition at line 306 of file LinearStatistic.c.

References C_ExpectCovarLinearStatistic(), ncol(), nrow(), PL2_covarianceSym, and PL2_expectationSym.

Here is the call graph for this function:



5.9.2.7 SEXP R_LinearStatistic (SEXP *x*, SEXP *y*, SEXP *weights*)

R-interface to C_LinearStatistic

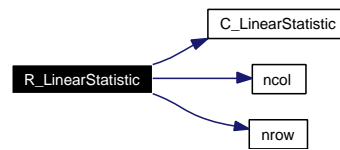
Parameters:

- x* values of the transformation
- y* values of the influence function
- weights* case weights

Definition at line 59 of file LinearStatistic.c.

References C_LinearStatistic(), ncol(), and nrow().

Here is the call graph for this function:



5.9.2.8 SEXP R_PermutedLinearStatistic (SEXP *x*, SEXP *y*, SEXP *indx*, SEXP *perm*)

Linear Statistic with permuted indices

Parameters:

- x* values of the transformation
- y* values of the influence function
- indx* indices for the x-part
- perm* (permuted) indices for the y-part

Definition at line 384 of file LinearStatistic.c.

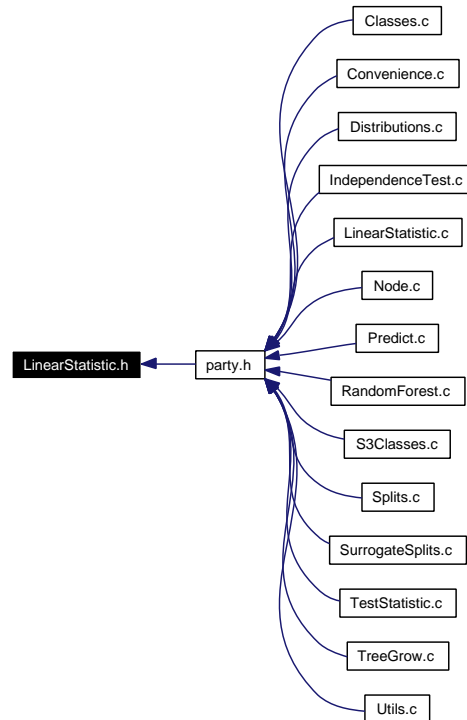
References `nrow()`.

Here is the call graph for this function:



5.10 LinearStatistic.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_LinearStatistic](#) (const double *x, const int p, const double *y, const int q, const double *weights, const int n, double *ans)
- void [C_ExpectCovarInfluence](#) (const double *y, const int q, const double *weights, const int n, SEXP ans)
- void [C_ExpectCovarLinearStatistic](#) (const double *x, const int p, const double *y, const int q, const double *weights, const int n, const SEXP expcovinf, SEXP ans)
- void [C_PermutedLinearStatistic](#) (const double *x, const int p, const double *y, const int q, const int n, const int nperm, const int *indx, const int *perm, double *ans)
- SEXP [R_ExpectCovarInfluence](#) (SEXP y, SEXP weights)

5.10.1 Function Documentation

5.10.1.1 void [C_ExpectCovarInfluence](#) (const double *y, const int q, const double *weights, const int n, SEXP ans)

Conditional expectation and covariance of the influence function

Parameters:

- y* values of the influence function
- q* dimension of the influence function

weights case weights
n number of observations
ans return value; an object of class 'ExpectCovarInfluence'

Definition at line 101 of file LinearStatistic.c.

References PL2_covarianceSym, PL2_expectationSym, and PL2_sumweightsSym.

Referenced by C_GlobalTest(), C_LinStatExpCov(), and R_ExpectCovarInfluence().

5.10.1.2 void C_ExpectCovarLinearStatistic (const double * *x*, const int *p*, const double * *y*, const int *q*, const double * *weights*, const int *n*, const SEXP *expcovinf*, SEXP *ans*)

Conditional expectation and covariance of the a linear statistic

Parameters:

x values of the transformation
p dimension of the transformation
y values of the influence function
q dimension of the influence function
weights case weights
n number of observations
expcovinf an object of class 'ExpectCovarInfluence'
ans return value; an object of class 'ExpectCovar'

Definition at line 213 of file LinearStatistic.c.

References PL2_covarianceSym, PL2_expectationSym, and PL2_sumweightsSym.

Referenced by C_LinStatExpCov(), and R_ExpectCovarLinearStatistic().

5.10.1.3 void C_LinearStatistic (const double * *x*, const int *p*, const double * *y*, const int *q*, const double * *weights*, const int *n*, double * *ans*)

Computes the linear statistic, formula (1) in the paper

Parameters:

x values of the transformation
p dimension of the transformation
y values of the influence function
q dimension of the influence function
weights case weights
n number of observations
ans return value; a pointer to a REALSXP-vector of length pq

Definition at line 23 of file LinearStatistic.c.

Referenced by C_LinStatExpCov(), and R_LinearStatistic().

5.10.1.4 void C_PermutedLinearStatistic (const double * *x*, const int *p*, const double * *y*, const int *q*, const int *n*, const int *nperm*, const int * *indx*, const int * *perm*, double * *ans*)

Linear Statistic with permuted indices

Parameters:

- x* values of the transformation
- p* dimension of the transformation
- y* values of the influence function
- q* dimension of the influence function
- n* number of observations
- nperm* number of permutations
- indx* indices for the x-part
- perm* (permuted) indices for the y-part
- ans* return value; a pointer to a REALSXP-vector of length pq

Definition at line 351 of file LinearStatistic.c.

5.10.1.5 SEXP R_ExpectCovarInfluence (SEXP *y*, SEXP *weights*)

R-interface to C_ExpectCovarInfluence

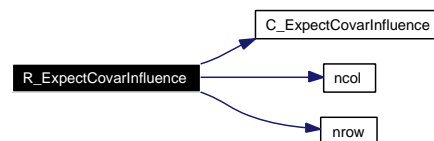
Parameters:

- y* values of the influence function
- weights* case weights

Definition at line 171 of file LinearStatistic.c.

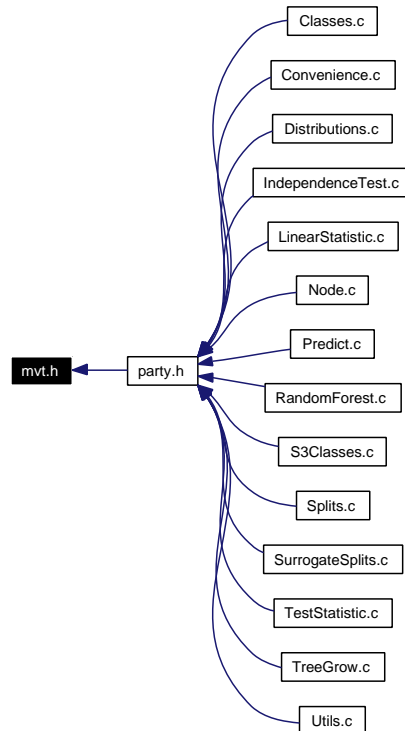
References C_ExpectCovarInfluence(), ncol(), nrow(), PL2_covarianceSym, PL2_expectationSym, and PL2_sumweightsSym.

Here is the call graph for this function:



5.11 mvt.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void F77_NAME() [mvtdst](#) (int *n, int *nu, double *lower, double *upper, int *infin, double *corr, double *delta, int *maxpts, double *abseps, double *releps, double *error, double *value, int *inform)

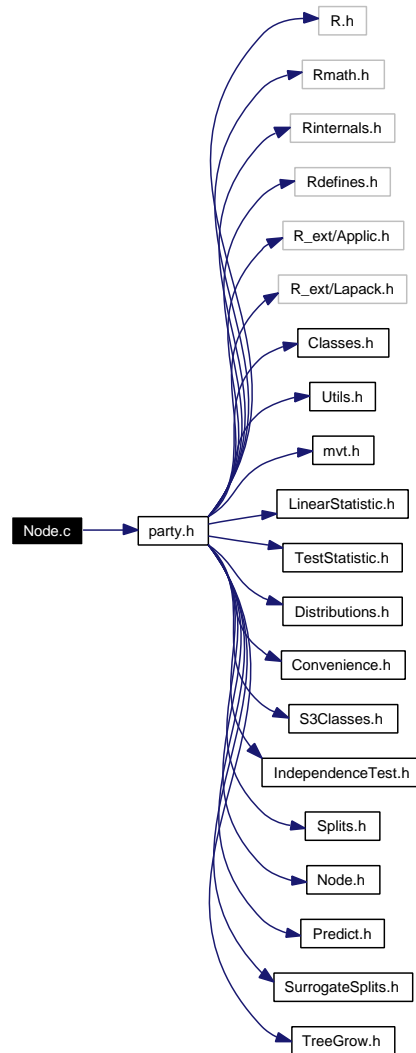
5.11.1 Function Documentation

- 5.11.1.1** void F77_NAME() [mvtdst](#) (int * *n*, int * *nu*, double * *lower*, double * *upper*, int * *infin*, double * *corr*, double * *delta*, int * *maxpts*, double * *abseps*, double * *releps*, double * *error*, double * *value*, int * *inform*)

5.12 Node.c File Reference

```
#include "party.h"
```

Include dependency graph for Node.c:



Functions

- void [C_prediction](#) (const double *y, int n, int q, const double *weights, const double sweights, double *ans)
- void [C_Node](#) (SEXP node, SEXP learnsample, SEXP weights, SEXP fitmem, SEXP controls, int TERMINAL)
- SEXP [R_Node](#) (SEXP learnsample, SEXP weights, SEXP fitmem, SEXP controls)

5.12.1 Detailed Description

Node computations

Author

hothorn

Date

2007-07-23 10:09:38 +0200 (Mon, 23 Jul 2007)

Definition in file [Node.c](#).**5.12.2 Function Documentation****5.12.2.1 void C_Node (SEXP *node*, SEXP *learnsample*, SEXP *weights*, SEXP *fitmem*, SEXP *controls*, int *TERMINAL*)**

The main function for all node computations

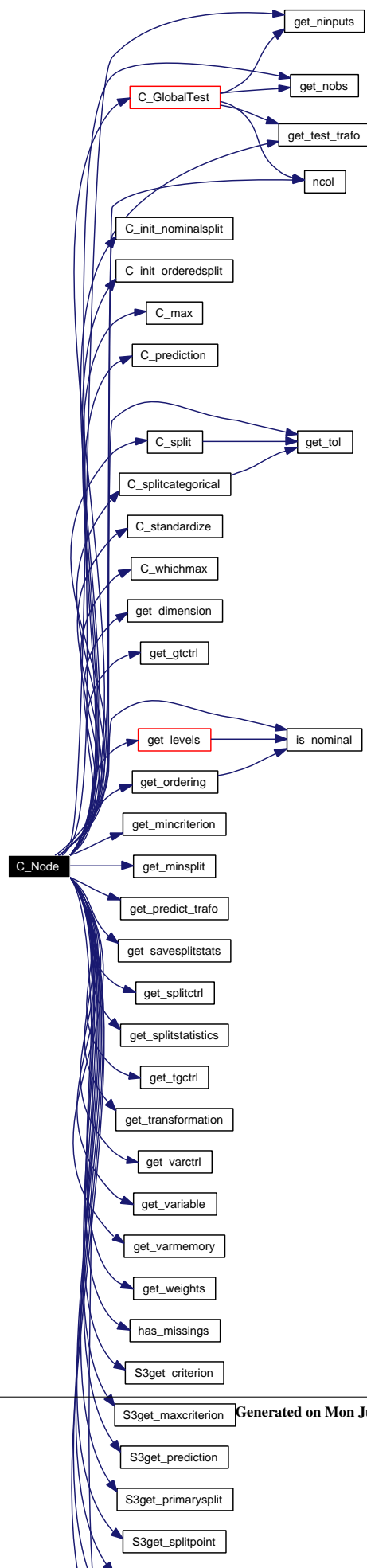
Parameters:*node* an initialized node (an S3 object!)*learnsample* an object of class ‘LearningSample’*weights* case weights*fitmem* an object of class ‘TreeFitMemory’*controls* an object of class ‘TreeControl’*TERMINAL* logical indicating if this node will be a terminal node

Definition at line 48 of file Node.c.

References C_GlobalTest(), C_init_nominalsplit(), C_init_orderedsplit(), C_max(), C_prediction(), C_split(), C_splitcategorical(), C_standardize(), C_whichmax(), get_dimension(), get_gtctrl(), get_levels(), get_mincriterion(), get_minsplit(), get_ninputs(), get_nobs(), get_ordering(), get_predict_trafo(), get_savesplitstats(), get_splitctrl(), get_splitstatistics(), get_test_trafo(), get_tgctrl(), get_tol(), get_transformation(), get_varctrl(), get_variable(), get_varmemory(), get_weights(), has_missings(), is_nominal(), ncol(), PL2_covarianceSym, PL2_expcovinfSym, PL2_expectationSym, PL2_inputsSym, PL2_linearstatisticSym, PL2_linexpcov2sampleSym, PL2_responsesSym, PL2_sumweightsSym, S3_SUMWEIGHTS, S3get_criterion(), S3get_maxcriterion(), S3get_prediction(), S3get_primarysplit(), S3get_splitpoint(), S3get_splitstatistics(), S3get_table(), S3get_teststat(), and S3set_variableID().

Referenced by C_TreeGrow(), and R_Node().

Here is the call graph for this function:



5.12.2.2 void C_prediction (const double * *y*, int *n*, int *q*, const double * *weights*, const double * *sweights*, double * *ans*)

Compute prediction of a node

Parameters:

- y* the response variable (raw numeric values or dummy encoded factor)
- n* number of observations
- q* number of columns of *y*
- weights* case weights
- sweights* sum of case weights
- ans* return value; the *q*-dimensional predictions

Definition at line 22 of file Node.c.

Referenced by C_Node().

5.12.2.3 SEXP R_Node (SEXP *learnsample*, SEXP *weights*, SEXP *fitmem*, SEXP *controls*)

R-interface to C_Node

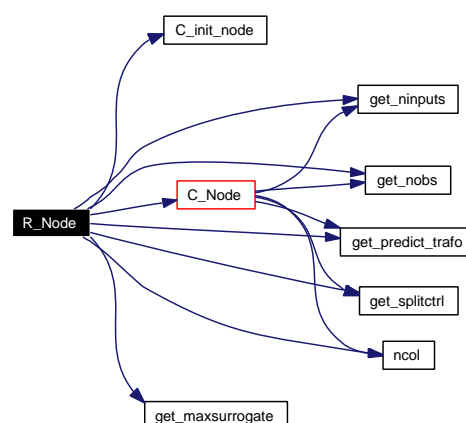
Parameters:

- learnsample* an object of class ‘LearningSample’
- weights* case weights
- fitmem* an object of class ‘TreeFitMemory’
- controls* an object of class ‘TreeControl’

Definition at line 228 of file Node.c.

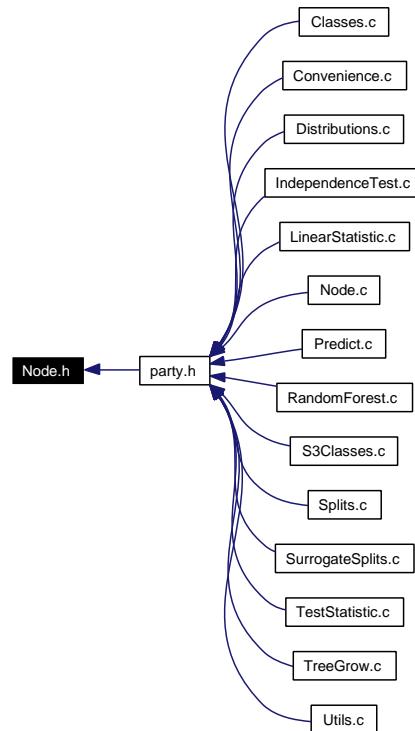
References C_init_node(), C_Node(), get_maxsurrogate(), get_ninputs(), get_nobs(), get_predict_trafo(), get_splitctrl(), ncol(), NODE_LENGTH, and PL2_responsesSym.

Here is the call graph for this function:



5.13 Node.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_Node](#) (SEXP node, SEXP learnsample, SEXP weights, SEXP fitmem, SEXP controls, int TERMINAL)

5.13.1 Function Documentation

5.13.1.1 void C_Node (SEXP *node*, SEXP *learnsample*, SEXP *weights*, SEXP *fitmem*, SEXP *controls*, int *TERMINAL*)

The main function for all node computations

Parameters:

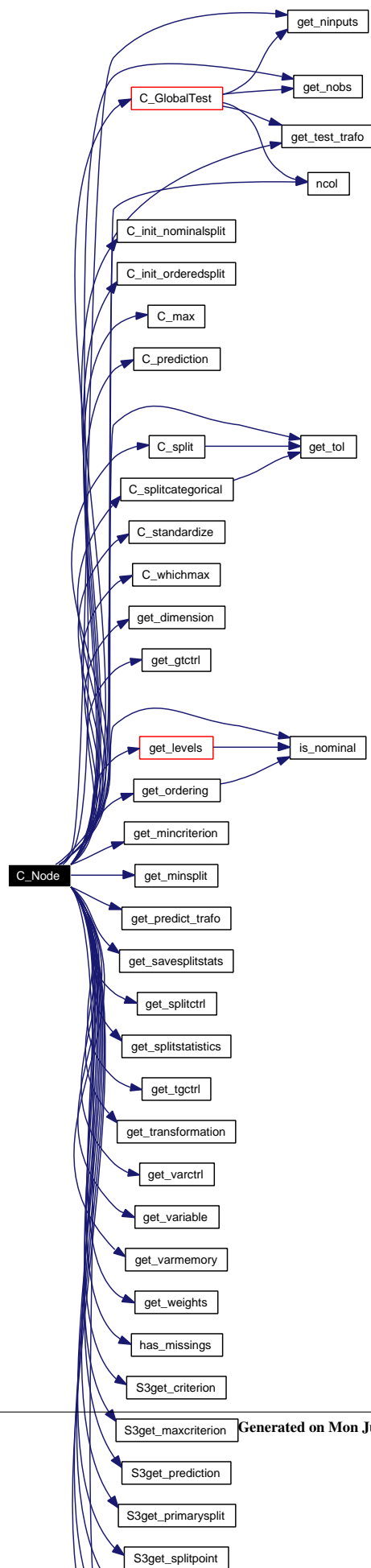
- node* an initialized node (an S3 object!)
- learnsample* an object of class ‘LearningSample’
- weights* case weights
- fitmem* an object of class ‘TreeFitMemory’
- controls* an object of class ‘TreeControl’
- TERMINAL* logical indicating if this node will be a terminal node

Definition at line 48 of file Node.c.

References C_GlobalTest(), C_init_nominalsplit(), C_init_orderedsplit(), C_max(), C_prediction(), C_split(), C_splitcategorical(), C_standardize(), C_whichmax(), get_dimension(), get_gtctrl(), get_levels(), get_mincriterion(), get_minsplit(), get_ninputs(), get_nobs(), get_ordering(), get_predict_trafo(), get_savesplitstats(), get_splitctrl(), get_splitstatistics(), get_test_trafo(), get_tgctrl(), get_tol(), get_transformation(), get_varctrl(), get_variable(), get_varmemory(), get_weights(), has_missings(), is_nominal(), ncol(), PL2_covarianceSym, PL2_expcovinfSym, PL2_expectationSym, PL2_inputsSym, PL2_linearstatisticSym, PL2_linexpcov2sampleSym, PL2_responsesSym, PL2_sumweightsSym, S3_SUMWEIGHTS, S3get_criterion(), S3get_maxcriterion(), S3get_prediction(), S3get_primarysplit(), S3get_splitpoint(), S3get_splitstatistics(), S3get_table(), S3get_teststat(), and S3set_variableID().

Referenced by C_TreeGrow(), and R_Node().

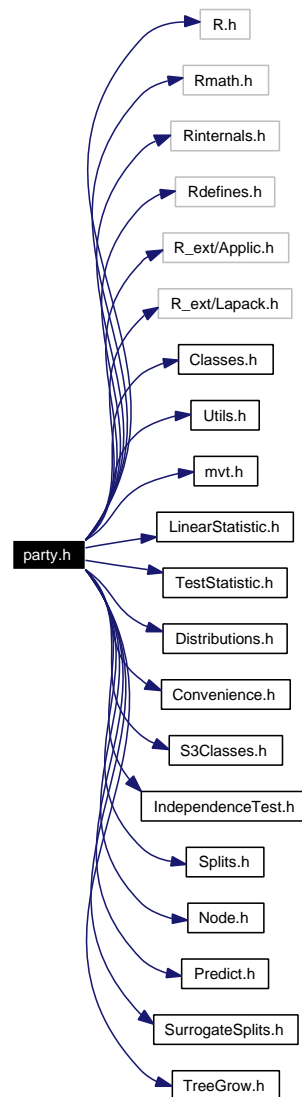
Here is the call graph for this function:



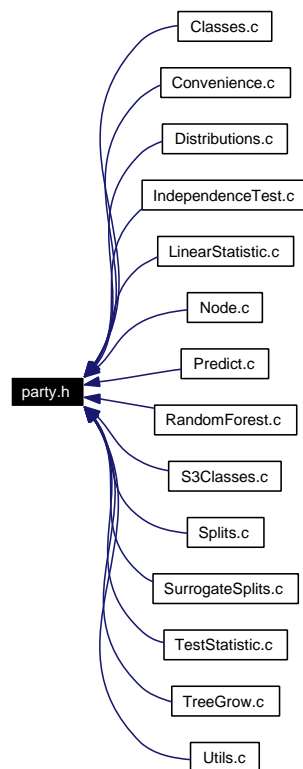
5.14 party.h File Reference

```
#include <R.h>
#include <Rmath.h>
#include <Rinternals.h>
#include <Rdefines.h>
#include <R_ext/Applic.h>
#include <R_ext/Lapack.h>
#include "Classes.h"
#include "Utils.h"
#include "mvt.h"
#include "LinearStatistic.h"
#include "TestStatistic.h"
#include "Distributions.h"
#include "Convenience.h"
#include "S3Classes.h"
#include "IndependenceTest.h"
#include "Splits.h"
#include "Node.h"
#include "Predict.h"
#include "SurrogateSplits.h"
#include "TreeGrow.h"
```

Include dependency graph for party.h:



This graph shows which files directly or indirectly include this file:



Defines

- #define [S3_NODEID](#) 0
- #define [S3_WEIGHTS](#) 1
- #define [S3_CRITERION](#) 2
- #define [S3_TERMINAL](#) 3
- #define [S3_PSPLIT](#) 4
- #define [S3_SSPLIT](#) 5
- #define [S3_PREDICTION](#) 6
- #define [S3_LEFT](#) 7
- #define [S3_RIGHT](#) 8
- #define [S3_SUMWEIGHTS](#) 9
- #define [NODE_LENGTH](#) 10
- #define [S3_STATISTICS](#) 0
- #define [S3_iCRITERION](#) 1
- #define [S3_MAXCRITERION](#) 2
- #define [CRITERION_LENGTH](#) 3
- #define [S3_VARIABLEID](#) 0
- #define [S3_ORDERED](#) 1
- #define [S3_SPLITPOINT](#) 2
- #define [S3_SPLITSTATISTICS](#) 3
- #define [S3_TOLEFT](#) 4
- #define [S3_TABLE](#) 5
- #define [SPLIT_LENGTH](#) 6
- #define [MAXABS](#) 1

- #define QUADFORM 2
- #define BONFERRONI 1
- #define MONTECARLO 2
- #define AGGREGATED 3
- #define UNIVARIATE 4
- #define TESTSTATISTIC 5

5.14.1 Define Documentation

5.14.1.1 #define AGGREGATED 3

Definition at line 68 of file party.h.

5.14.1.2 #define BONFERRONI 1

Definition at line 66 of file party.h.

5.14.1.3 #define CRITERION_LENGTH 3

Definition at line 50 of file party.h.

Referenced by C_init_node().

5.14.1.4 #define MAXABS 1

Definition at line 62 of file party.h.

Referenced by C_ConditionalPvalue().

5.14.1.5 #define MONTECARLO 2

Definition at line 67 of file party.h.

5.14.1.6 #define NODE_LENGTH 10

Definition at line 44 of file party.h.

Referenced by C_init_node(), C_splitnode(), R_Node(), and R_TreeGrow().

5.14.1.7 #define QUADFORM 2

Definition at line 63 of file party.h.

Referenced by C_ConditionalPvalue().

5.14.1.8 #define S3_CRITERION 2

Definition at line 36 of file party.h.

Referenced by C_init_node(), S3get_criterion(), S3get_maxcriterion(), and S3get_teststat().

5.14.1.9 #define S3_iCRITERION 1

Definition at line 48 of file party.h.

Referenced by C_init_node(), and S3get_criterion().

5.14.1.10 #define S3_LEFT 7

Definition at line 41 of file party.h.

Referenced by C_splitnode(), and S3get_leftnode().

5.14.1.11 #define S3_MAXCRITERION 2

Definition at line 49 of file party.h.

Referenced by C_init_node(), and S3get_maxcriterion().

5.14.1.12 #define S3_NODEID 0

Definition at line 34 of file party.h.

Referenced by C_init_node(), S3get_nodeID(), and S3set_nodeID().

5.14.1.13 #define S3_ORDERED 1

Definition at line 54 of file party.h.

Referenced by C_init_nominalsplit(), C_init_orderedsplit(), S3is_ordered(), S3set_nominal(), and S3set_ordered().

5.14.1.14 #define S3_PREDICTION 6

Definition at line 40 of file party.h.

Referenced by C_init_node(), and S3get_prediction().

5.14.1.15 #define S3_PSPLIT 4

Definition at line 38 of file party.h.

Referenced by C_init_node(), and S3get_primarysplit().

5.14.1.16 #define S3_RIGHT 8

Definition at line 42 of file party.h.

Referenced by C_splitnode(), and S3get_rightnode().

5.14.1.17 #define S3_SPLITPOINT 2

Definition at line 55 of file party.h.

Referenced by C_init_nominalsplit(), C_init_orderedsplit(), and S3get_splitpoint().

5.14.1.18 #define S3_SPLITSTATISTICS 3

Definition at line 56 of file party.h.

Referenced by C_init_nominalsplit(), C_init_orderedsplit(), and S3get_splitstatistics().

5.14.1.19 #define S3_SSPLIT 5

Definition at line 39 of file party.h.

Referenced by C_init_node(), and S3get_surrogatesplits().

5.14.1.20 #define S3_STATISTICS 0

Definition at line 47 of file party.h.

Referenced by C_init_node(), and S3get_teststat().

5.14.1.21 #define S3_SUMWEIGHTS 9

Definition at line 43 of file party.h.

Referenced by C_init_node(), C_Node(), and S3get_sumweights().

5.14.1.22 #define S3_TABLE 5

Definition at line 58 of file party.h.

Referenced by C_init_nominalsplit(), C_init_orderedsplit(), and S3get_table().

5.14.1.23 #define S3_TERMINAL 3

Definition at line 37 of file party.h.

Referenced by C_init_node(), S3get_nodeterminal(), and S3set_nodeterminal().

5.14.1.24 #define S3_TOLEFT 4

Definition at line 57 of file party.h.

Referenced by C_init_nominalsplit(), C_init_orderedsplit(), S3get_toleft(), and S3set_toleft().

5.14.1.25 #define S3_VARIABLEID 0

Definition at line 53 of file party.h.

Referenced by C_init_nominalsplit(), C_init_orderedsplit(), S3get_variableID(), and S3set_variableID().

5.14.1.26 #define S3_WEIGHTS 1

Definition at line 35 of file party.h.

Referenced by C_init_node(), C_remove_weights(), and S3get_nodeweights().

5.14.1.27 #define SPLIT_LENGTH 6

Definition at line 59 of file party.h.

Referenced by C_init_node(), C_init_nominalsplit(), and C_init_orderedsplit().

5.14.1.28 #define TESTSTATISTIC 5

Definition at line 70 of file party.h.

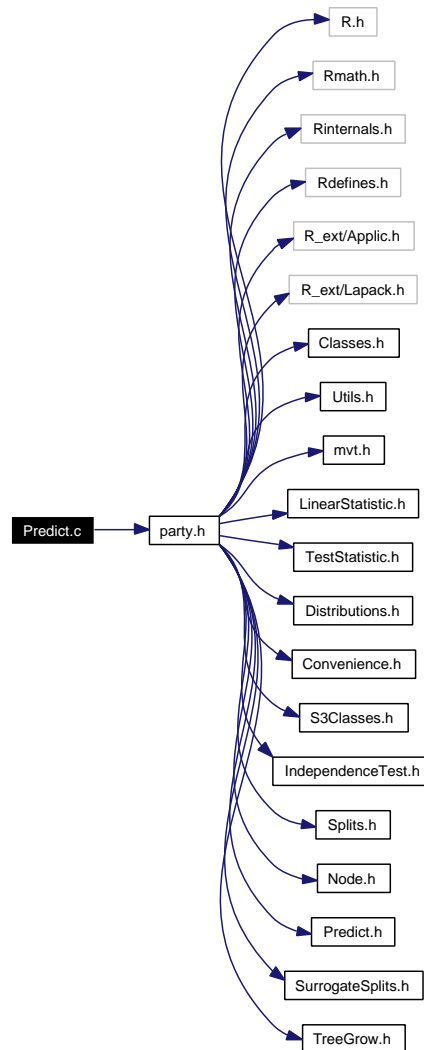
5.14.1.29 #define UNIVARIATE 4

Definition at line 69 of file party.h.

5.15 Predict.c File Reference

```
#include "party.h"
```

Include dependency graph for Predict.c:



Functions

- void [C_splitnode](#) (SEXP node, SEXP learnsample, SEXP control)
- SEXP [C_get_node](#) (SEXP subtree, SEXP newinputs, double mincriterion, int numobs)
- SEXP [R_get_node](#) (SEXP subtree, SEXP newinputs, SEXP mincriterion, SEXP numobs)
- SEXP [C_get_nodebynum](#) (SEXP subtree, int nodenum)
- SEXP [R_get_nodebynum](#) (SEXP subtree, SEXP nodenum)
- SEXP [C_get_prediction](#) (SEXP subtree, SEXP newinputs, double mincriterion, int numobs)
- SEXP [C_get_nodeweights](#) (SEXP subtree, SEXP newinputs, double mincriterion, int numobs)
- int [C_get_nodeID](#) (SEXP subtree, SEXP newinputs, double mincriterion, int numobs)
- SEXP [R_get_nodeID](#) (SEXP tree, SEXP newinputs, SEXP mincriterion)
- void [C_predict](#) (SEXP tree, SEXP newinputs, double mincriterion, SEXP ans)

- SEXPR [R_predict](#) (SEXP tree, SEXP newinputs, SEXP mincriterion)
- void [C_getpredictions](#) (SEXP tree, SEXP where, SEXP ans)
- SEXPR [R_getpredictions](#) (SEXP tree, SEXP where)
- SEXPR [R_predictRF_weights](#) (SEXP forest, SEXP where, SEXP weights, SEXP newinputs, SEXP mincriterion, SEXP oobpred)

5.15.1 Detailed Description

Node splitting and prediction

Author

hothorn

Date

2007-07-23 10:02:09 +0200 (Mon, 23 Jul 2007)

Definition in file [Predict.c](#).

5.15.2 Function Documentation

5.15.2.1 SEXPR [C_get_node](#) (SEXP *subtree*, SEXP *newinputs*, double *mincriterion*, int *numobs*)

Get the terminal node for obs. number ‘numobs’ of ‘newinputs’

Parameters:

subtree a tree

newinputs an object of class ‘VariableFrame’

mincriterion overwrites mincriterion used for tree growing

numobs observation number

Todo

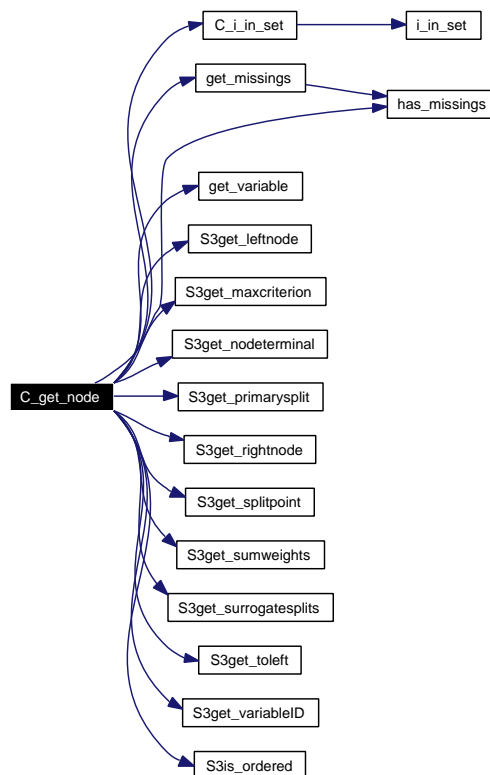
handle surrogate splits

Definition at line 120 of file Predict.c.

References [C_i_in_set\(\)](#), [get_missings\(\)](#), [get_variable\(\)](#), [has_missings\(\)](#), [S3get_leftnode\(\)](#), [S3get_maxcriterion\(\)](#), [S3get_nodeterminal\(\)](#), [S3get_primarysplit\(\)](#), [S3get_rightnode\(\)](#), [S3get_splitpoint\(\)](#), [S3get_sumweights\(\)](#), [S3get_surrogatesplits\(\)](#), [S3get_toleft\(\)](#), [S3get_variableID\(\)](#), and [S3is_ordered\(\)](#).

Referenced by [C_get_nodeID\(\)](#), [C_get_nodeweights\(\)](#), [C_get_prediction\(\)](#), and [R_get_node\(\)](#).

Here is the call graph for this function:



5.15.2.2 SEXP C_get_nodebynum (SEXP subtree, int nodenum)

Get the node with nodeID 'nodenum'

Parameters:

subtree a tree

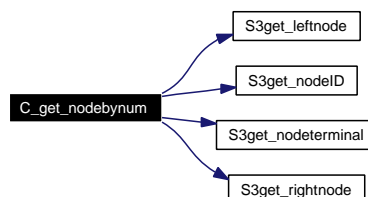
nodenum a nodeID

Definition at line 242 of file Predict.c.

References `S3get_leftnode()`, `S3get_nodeID()`, `S3get_nodeterminal()`, and `S3get_rightnode()`.

Referenced by `C_getpredictions()`, `R_get_nodebynum()`, and `R_predictRF_weights()`.

Here is the call graph for this function:



5.15.2.3 int C_get_nodeID (SEXP *subtree*, SEXP *newinputs*, double *mincriterion*, int *numobs*)

Get the nodeID for a new observation

Parameters:

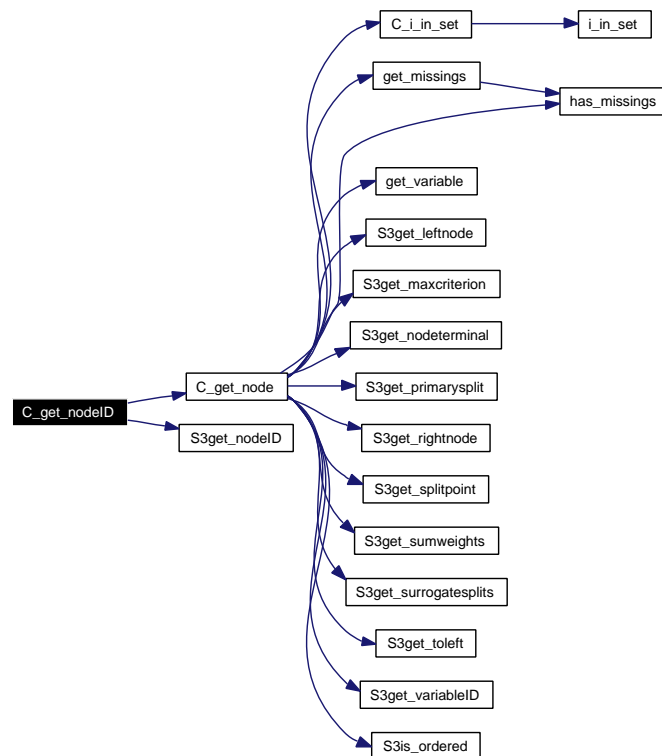
- subtree* a tree
- newinputs* an object of class 'VariableFrame'
- mincriterion* overwrites mincriterion used for tree growing
- numobs* observation number

Definition at line 306 of file Predict.c.

References C_get_node(), and S3get_nodeID().

Referenced by R_get_nodeID(), and R_predictRF_weights().

Here is the call graph for this function:



5.15.2.4 SEXP C_get_nodeweights (SEXP *subtree*, SEXP *newinputs*, double *mincriterion*, int *numobs*)

Get the weights for a new observation

Parameters:

- subtree* a tree
- newinputs* an object of class 'VariableFrame'

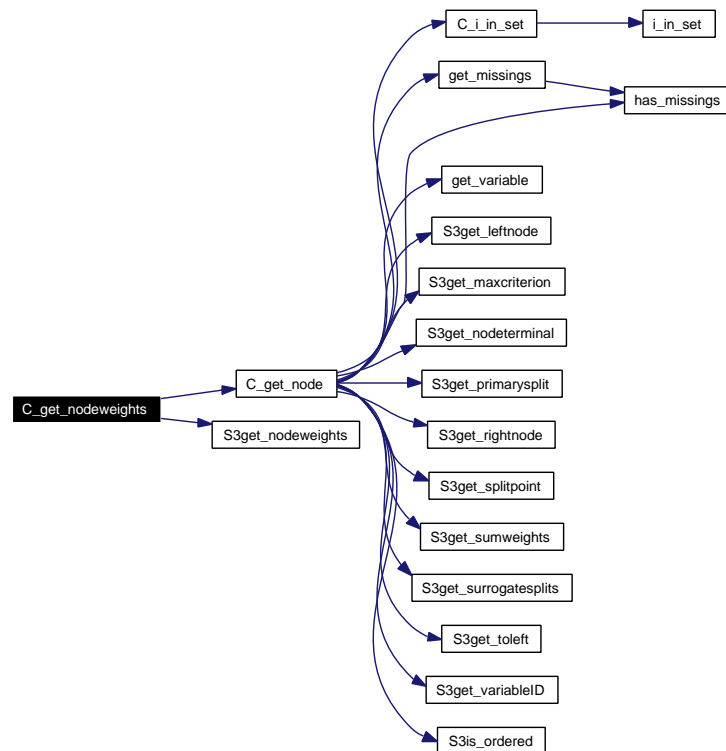
mincriterion overwrites mincriterion used for tree growing

numobs observation number

Definition at line 291 of file Predict.c.

References C_get_node(), and S3get_nodeweights().

Here is the call graph for this function:



5.15.2.5 SEXP C_get_prediction (SEXP *subtree*, SEXP *newinputs*, double *mincriterion*, int *numobs*)

Get the prediction of a new observation

Parameters:

subtree a tree

newinputs an object of class ‘VariableFrame’

mincriterion overwrites mincriterion used for tree growing

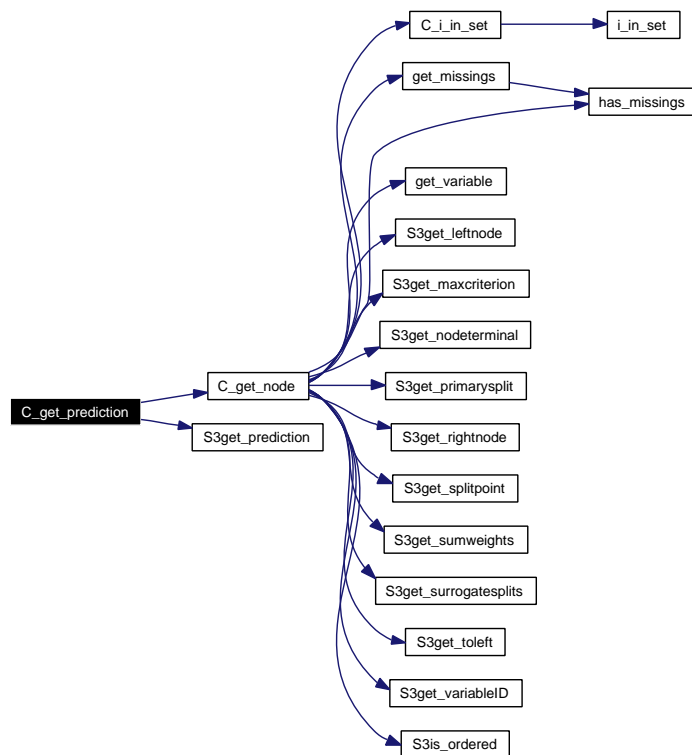
numobs observation number

Definition at line 276 of file Predict.c.

References C_get_node(), and S3get_prediction().

Referenced by C_predict().

Here is the call graph for this function:



5.15.2.6 void C_getpredictions (SEXP *tree*, SEXP *where*, SEXP *ans*)

Get the predictions from ‘where’ nodes

Parameters:

tree a tree

where vector of nodeID’s

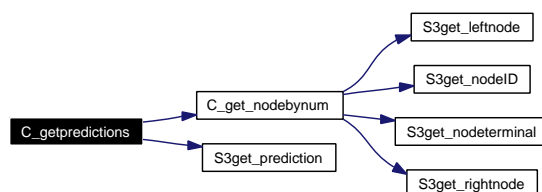
ans return value

Definition at line 384 of file Predict.c.

References C_get_nodebynum(), and S3get_prediction().

Referenced by R_getpredictions().

Here is the call graph for this function:



5.15.2.7 void C_predict (SEXP *tree*, SEXP *newinputs*, double *mincriterion*, SEXP *ans*)

Get all predictions for 'newinputs'

Parameters:

tree a tree

newinputs an object of class 'VariableFrame'

mincriterion overwrites mincriterion used for tree growing

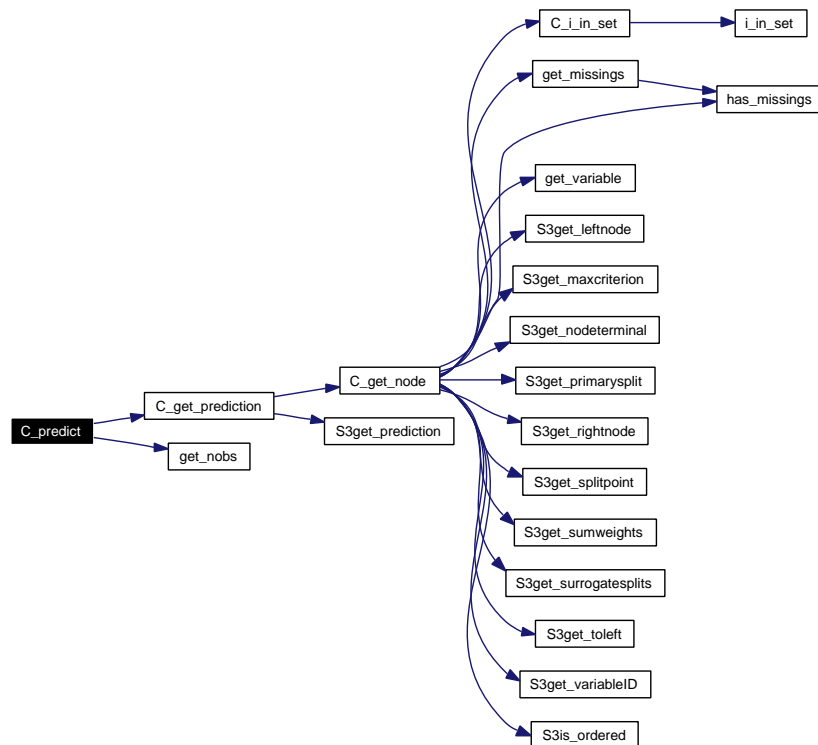
ans return value

Definition at line 343 of file Predict.c.

References C_get_prediction(), and get_nobs().

Referenced by R_predict().

Here is the call graph for this function:



5.15.2.8 void C_splitnode (SEXP *node*, SEXP *learnsample*, SEXP *control*)

Split a node according to a splitting rule

Parameters:

node the current node with primary split specified

learnsample learning sample

control an object of class 'TreeControl'

Todo

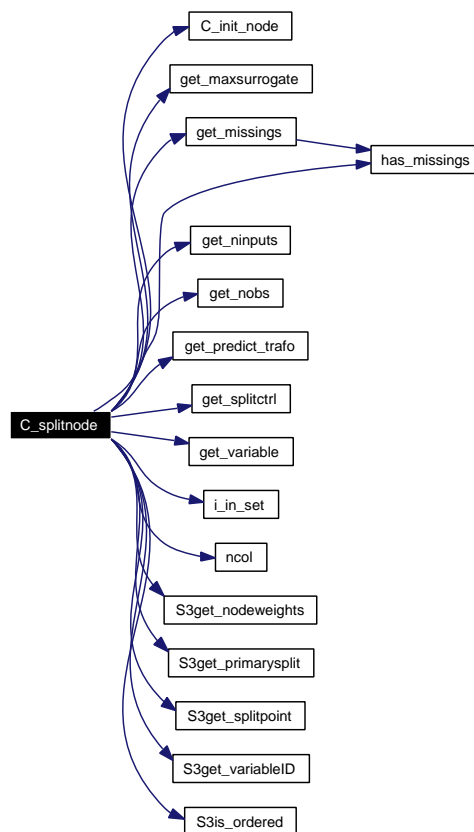
outplace the splitting since there are at least 3 functions with nearly identical code

Definition at line 21 of file Predict.c.

References C_init_node(), get_maxsurrogate(), get_missings(), get_ninputs(), get_nobs(), get_predict_trafo(), get_splitctrl(), get_variable(), has_missings(), i_in_set(), ncol(), NODE_LENGTH, PL2_inputsSym, PL2_responsesSym, S3_LEFT, S3_RIGHT, S3get_nodeweights(), S3get_primarysplit(), S3get_splitpoint(), S3get_variableID(), and S3is_ordered().

Referenced by C_TreeGrow().

Here is the call graph for this function:



5.15.2.9 SEXP R_get_node (SEXP subtree, SEXP newinputs, SEXP mincriterion, SEXP numobs)

R-Interface to C_get_node

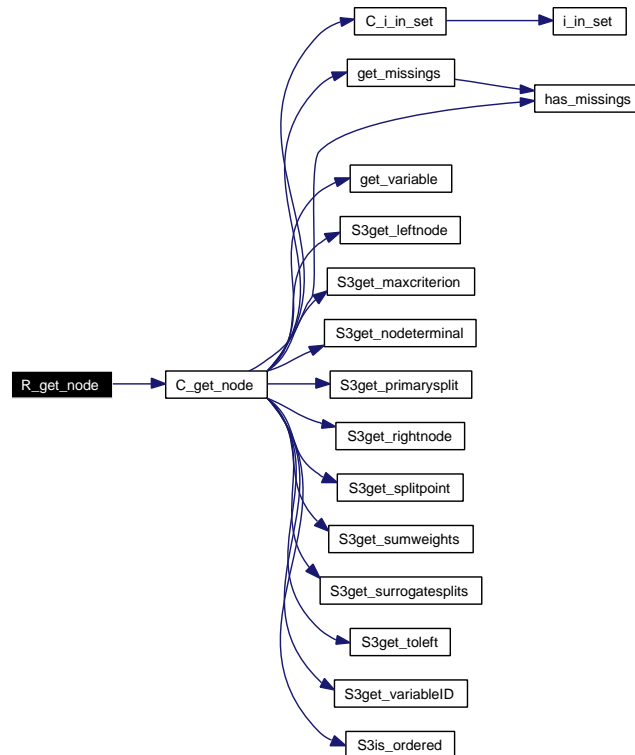
Parameters:

- subtree* a tree
- newinputs* an object of class ‘VariableFrame’
- mincriterion* overwrites mincriterion used for tree growing
- numobs* observation number

Definition at line 229 of file Predict.c.

References C_get_node().

Here is the call graph for this function:



5.15.2.10 SEXP R_get_nodebynum (SEXP subtree, SEXP nodenum)

R-Interface to C_get_nodenum

Parameters:

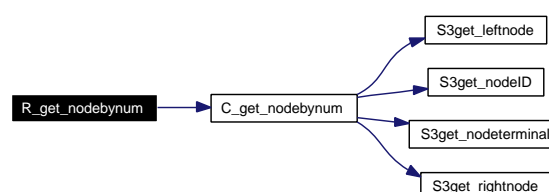
subtree a tree

nodenum a nodeID

Definition at line 263 of file Predict.c.

References C_get_nodebynum().

Here is the call graph for this function:



5.15.2.11 SEXP R_get_nodeID (SEXP *tree*, SEXP *newinputs*, SEXP *mincriterion*)

R-Interface to C_get_nodeID

Parameters:

tree a tree

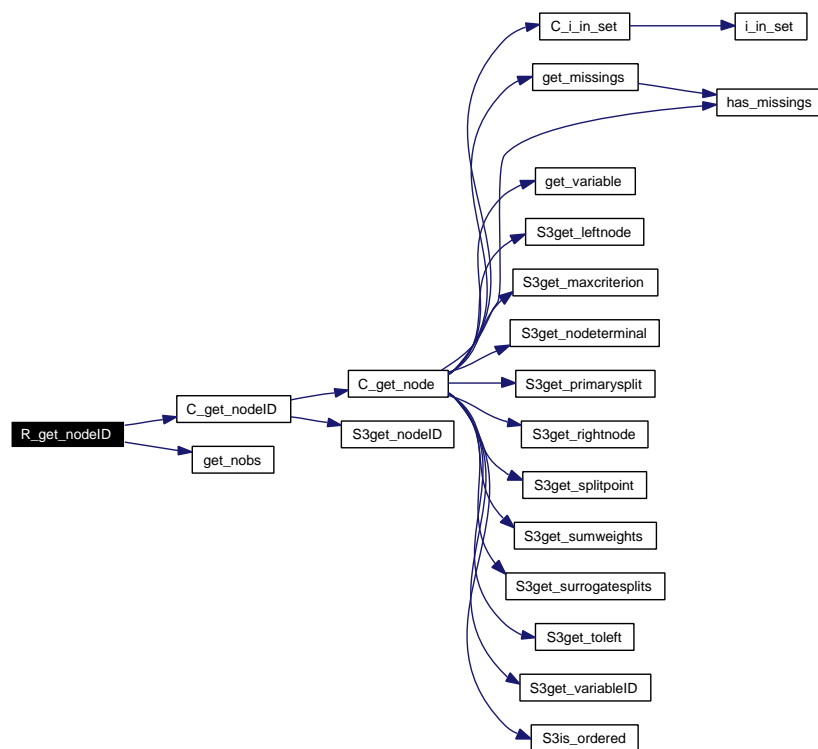
newinputs an object of class 'VariableFrame'

mincriterion overwrites mincriterion used for tree growing

Definition at line 320 of file Predict.c.

References C_get_nodeID(), and get_nobs().

Here is the call graph for this function:



5.15.2.12 SEXP R_getpredictions (SEXP *tree*, SEXP *where*)

R-Interface to C_getpredictions

Parameters:

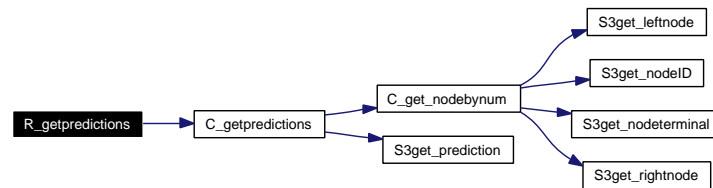
tree a tree

where vector of nodeID's

Definition at line 405 of file Predict.c.

References C_getpredictions().

Here is the call graph for this function:



5.15.2.13 SEXP R_predict (SEXP *tree*, SEXP *newinputs*, SEXP *mincriterion*)

R-Interface to C_predict

Parameters:

tree a tree

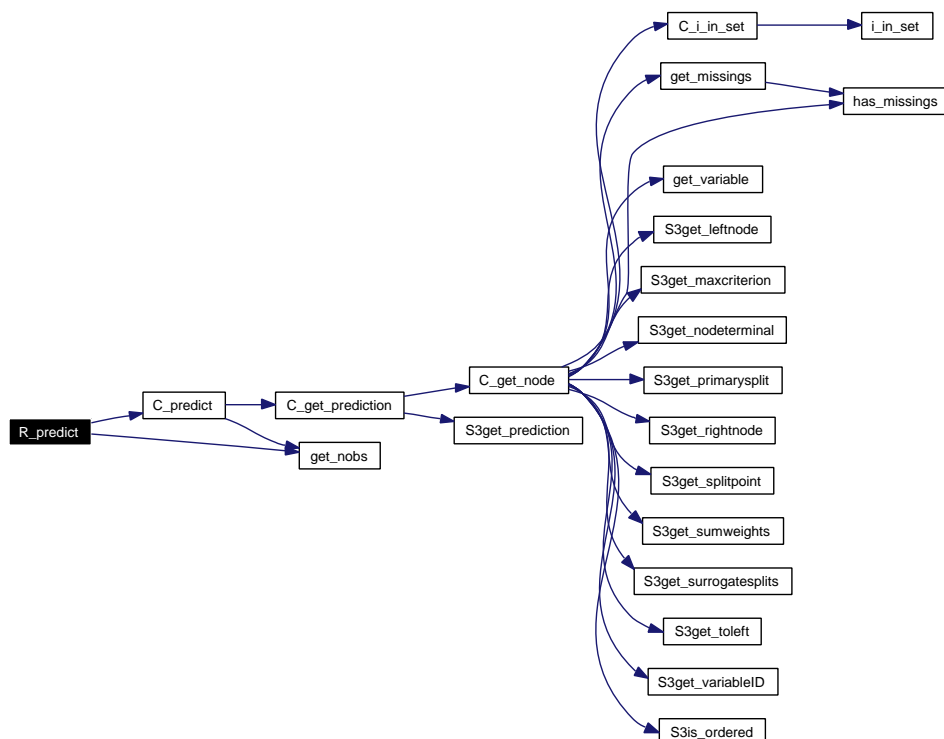
newinputs an object of class 'VariableFrame'

mincriterion overwrites mincriterion used for tree growing

Definition at line 364 of file Predict.c.

References C_predict(), and get_nobs().

Here is the call graph for this function:



5.15.2.14 SEXP R_predictRF_weights (SEXP forest, SEXP where, SEXP weights, SEXP newinputs, SEXP mincriterion, SEXP oobpred)

Predictions weights from RandomForest objects

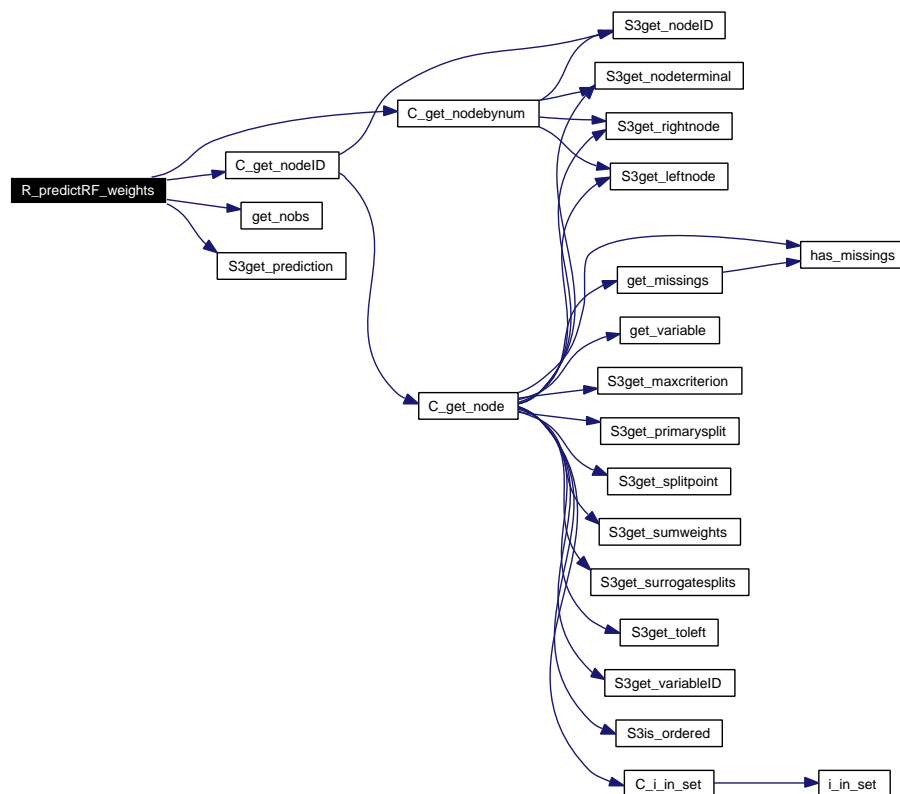
Parameters:

- forest* a list of trees
- where* integer matrix (n x ntree) for terminal node numbers
- weights* double matrix (n x ntree) for bootstrap case weights
- newinputs* an object of class 'VariableFrame'
- mincriterion* overwrites mincriterion used for tree growing
- oobpred* a logical indicating out-of-bag predictions

Definition at line 427 of file Predict.c.

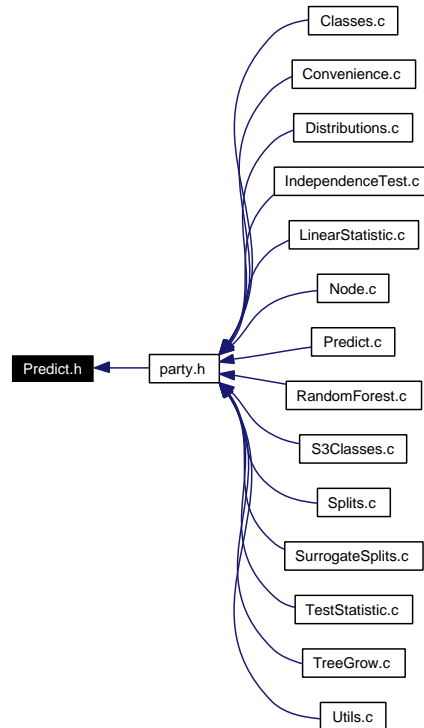
References C_get_nodebynum(), C_get_nodeID(), get_nobs(), and S3get_prediction().

Here is the call graph for this function:



5.16 Predict.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_splitnode](#) (SEXP node, SEXP learnsample, SEXP control)

5.16.1 Function Documentation

5.16.1.1 void C_splitnode (SEXP node, SEXP learnsample, SEXP control)

Split a node according to a splitting rule

Parameters:

- node* the current node with primary split specified
- learnsample* learning sample
- control* an object of class 'TreeControl'

Todo

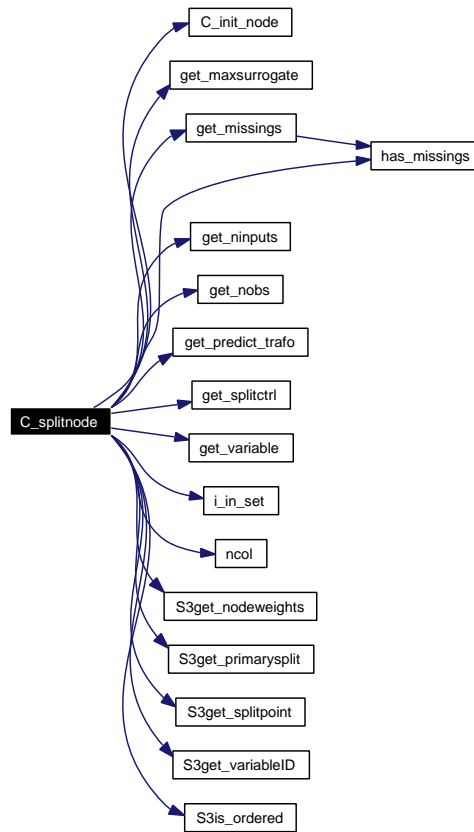
- outplace the splitting since there are at least 3 functions with nearly identical code

Definition at line 21 of file Predict.c.

References [C_init_node\(\)](#), [get_maxsurrogate\(\)](#), [get_missings\(\)](#), [get_ninputs\(\)](#), [get_nobs\(\)](#), [get_predict_trafo\(\)](#), [get_splitctrl\(\)](#), [get_variable\(\)](#), [has_missings\(\)](#), [i_in_set\(\)](#), [ncol\(\)](#), [NODE_LENGTH](#), [PL2_inputsSym](#), [PL2_responsesSym](#), [S3_LEFT](#), [S3_RIGHT](#), [S3get_nodeweights\(\)](#), [S3get_primarysplit\(\)](#), [S3get_splitpoint\(\)](#), [S3get_variableID\(\)](#), and [S3is_ordered\(\)](#).

Referenced by C_TreeGrow().

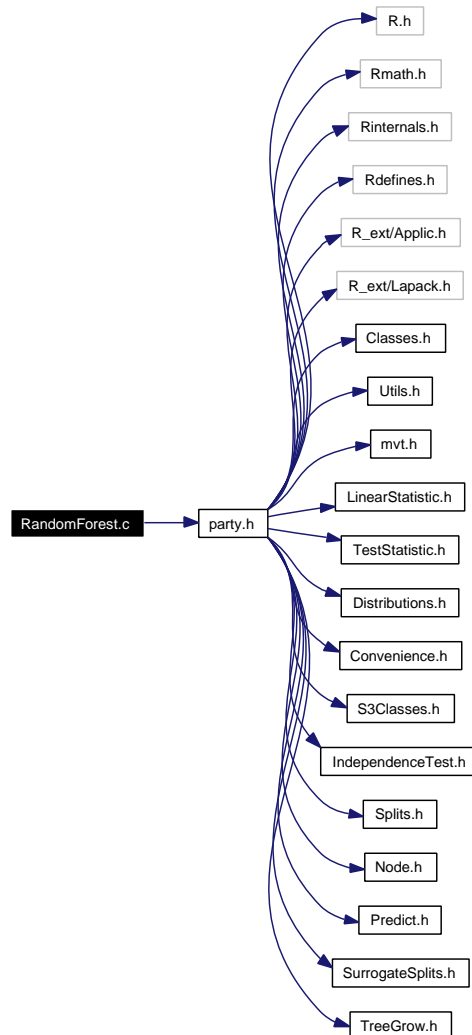
Here is the call graph for this function:



5.17 RandomForest.c File Reference

```
#include "party.h"
```

Include dependency graph for RandomForest.c:



Functions

- SEXP [R_Ensemble](#) (SEXP learnsample, SEXP weights, SEXP bwhere, SEXP bweights, SEXP fitmem, SEXP controls)

5.17.1 Detailed Description

Random forest with conditional inference trees

Author

hothorn

Date

2007-07-23 10:09:38 +0200 (Mon, 23 Jul 2007)

Definition in file [RandomForest.c](#).**5.17.2 Function Documentation****5.17.2.1 SEXP R_Ensemble (SEXP *learnsample*, SEXP *weights*, SEXP *bwhere*, SEXP *bweights*, SEXP *fitmem*, SEXP *controls*)**

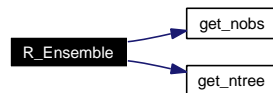
An experimental implementation of random forest like algorithms

Parameters:*learnsample* an object of class 'LearningSample'*weights* a vector of case weights*bwhere* integer matrix (n x ntree) for terminal node numbers*bweights* double matrix (n x ntree) for bootstrap case weights*fitmem* an object of class 'TreeFitMemory'*controls* an object of class 'TreeControl'

Definition at line 22 of file RandomForest.c.

References [get_nobs\(\)](#), and [get_ntree\(\)](#).

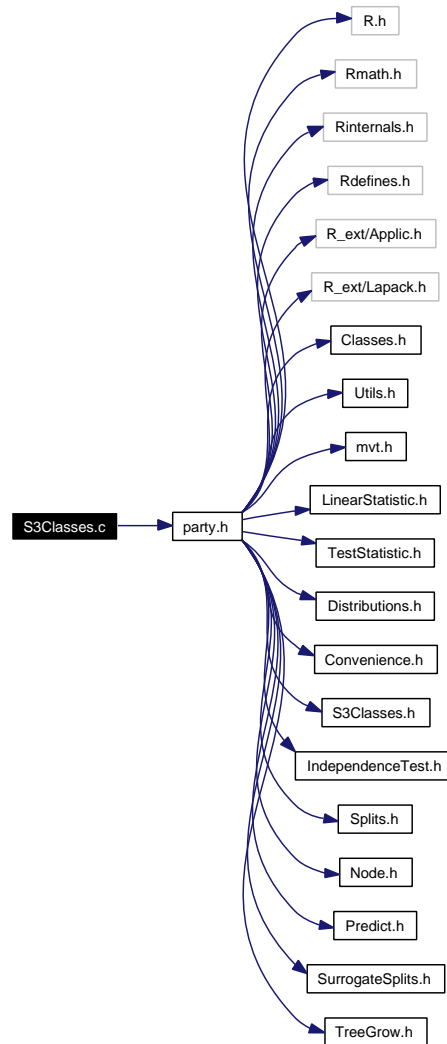
Here is the call graph for this function:



5.18 S3Classes.c File Reference

```
#include "party.h"
```

Include dependency graph for S3Classes.c:



Functions

- void [C_init_node](#) (SEXP node, int nobs, int ninputs, int nsurr, int q)
- void [S3set_nodeID](#) (SEXP node, int nodeID)
- int [S3get_nodeID](#) (SEXP node)
- SEXP [S3get_nodeweights](#) (SEXP node)
- double [S3get_sumweights](#) (SEXP node)
- SEXP [S3get_teststat](#) (SEXP node)
- SEXP [S3get_criterion](#) (SEXP node)
- SEXP [S3get_maxcriterion](#) (SEXP node)
- void [S3set_nodeterminal](#) (SEXP node)
- int [S3get_nodeterminal](#) (SEXP node)

- SEXP [S3get_primarysplit](#) (SEXP node)
- SEXP [S3get_surrogatesplits](#) (SEXP node)
- SEXP [S3get_prediction](#) (SEXP node)
- SEXP [S3get_leftnode](#) (SEXP node)
- SEXP [S3get_rightnode](#) (SEXP node)
- void [C_init_orderedsplit](#) (SEXP split, int nobs)
- void [C_init_nominalsplit](#) (SEXP split, int nlevels, int nobs)
- void [S3set_variableID](#) (SEXP split, int variableID)
- int [S3get_variableID](#) (SEXP split)
- int [S3is_ordered](#) (SEXP split)
- void [S3set_ordered](#) (SEXP split)
- void [S3set_nominal](#) (SEXP split)
- int [S3get_toleft](#) (SEXP split)
- void [S3set_toleft](#) (SEXP split, int left)
- SEXP [S3get_splitpoint](#) (SEXP split)
- SEXP [S3get_splitstatistics](#) (SEXP split)
- SEXP [S3get_table](#) (SEXP split)

5.18.1 Detailed Description

S3 classes for dealing with nodes and splits

Author

hothorn

Date

2007-07-23 09:44:00 +0200 (Mon, 23 Jul 2007)

Definition in file [S3Classes.c](#).

5.18.2 Function Documentation

5.18.2.1 void C_init_node (SEXP *node*, int *nobs*, int *ninputs*, int *nsurr*, int *q*)

Definition at line 11 of file S3Classes.c.

References [CRITERION_LENGTH](#), [NODE_LENGTH](#), [S3_CRITERION](#), [S3_iCRITERION](#), [S3_MAXCRITERION](#), [S3_NODEID](#), [S3_PREDICTION](#), [S3_PSPLIT](#), [S3_SSPLIT](#), [S3_STATISTICS](#), [S3_SUMWEIGHTS](#), [S3_TERMINAL](#), [S3_WEIGHTS](#), and [SPLIT_LENGTH](#).

Referenced by [C_splitnode\(\)](#), [R_Node\(\)](#), and [R_TreeGrow\(\)](#).

5.18.2.2 void C_init_nominalsplit (SEXP *split*, int *nlevels*, int *nobs*)

Definition at line 128 of file S3Classes.c.

References [S3_ORDERED](#), [S3_SPLITPOINT](#), [S3_SPLITSTATISTICS](#), [S3_TABLE](#), [S3_TOLEFT](#), [S3_VARIABLEID](#), and [SPLIT_LENGTH](#).

Referenced by [C_Node\(\)](#).

5.18.2.3 void C_init_orderedsplit (SEXP *split*, int *nobs*)

Definition at line 104 of file S3Classes.c.

References S3_ORDERED, S3_SPLITPOINT, S3_SPLITSTATISTICS, S3_TABLE, S3_TOLEFT, S3_VARIABLEID, and SPLIT_LENGTH.

Referenced by C_Node().

5.18.2.4 SEXP S3get_criterion (SEXP *node*)

Definition at line 68 of file S3Classes.c.

References S3_CRITERION, and S3_iCRITERION.

Referenced by C_Node().

5.18.2.5 SEXP S3get_leftnode (SEXP *node*)

Definition at line 96 of file S3Classes.c.

References S3_LEFT.

Referenced by C_get_node(), C_get_nodebynum(), C_remove_weights(), C_splitsurrogate(), and C_TreeGrow().

5.18.2.6 SEXP S3get_maxcriterion (SEXP *node*)

Definition at line 72 of file S3Classes.c.

References S3_CRITERION, and S3_MAXCRITERION.

Referenced by C_get_node(), and C_Node().

5.18.2.7 int S3get_nodeID (SEXP *node*)

Definition at line 47 of file S3Classes.c.

References S3_NODEID.

Referenced by C_get_nodebynum(), and C_get_nodeID().

5.18.2.8 int S3get_nodeterminal (SEXP *node*)

Definition at line 80 of file S3Classes.c.

References S3_TERMINAL.

Referenced by C_get_node(), C_get_nodebynum(), C_remove_weights(), and C_TreeGrow().

5.18.2.9 SEXP S3get_nodeweights (SEXP *node*)

Definition at line 51 of file S3Classes.c.

References S3_WEIGHTS.

Referenced by C_get_nodeweights(), C_splitnode(), C_splitsurrogate(), C_surrogates(), C_TreeGrow(), and R_TreeGrow().

5.18.2.10 SEXP S3get_prediction (SEXP *node*)

Definition at line 92 of file S3Classes.c.

References S3_PREDICTION.

Referenced by C_get_prediction(), C_getpredictions(), C_Node(), and R_predictRF_weights().

5.18.2.11 SEXP S3get_primarysplit (SEXP *node*)

Definition at line 84 of file S3Classes.c.

References S3_PSPLIT.

Referenced by C_get_node(), C_Node(), C_splitnode(), C_splitsurrogate(), and C_surrogates().

5.18.2.12 SEXP S3get_rightnode (SEXP *node*)

Definition at line 100 of file S3Classes.c.

References S3_RIGHT.

Referenced by C_get_node(), C_get_nodebynum(), C_remove_weights(), C_splitsurrogate(), and C_TreeGrow().

5.18.2.13 SEXP S3get_splitpoint (SEXP *split*)

Definition at line 179 of file S3Classes.c.

References S3_SPLITPOINT.

Referenced by C_get_node(), C_Node(), C_splitnode(), and C_splitsurrogate().

5.18.2.14 SEXP S3get_splitstatistics (SEXP *split*)

Definition at line 183 of file S3Classes.c.

References S3_SPLITSTATISTICS.

Referenced by C_Node().

5.18.2.15 double S3get_sumweights (SEXP *node*)

Definition at line 60 of file S3Classes.c.

References S3_SUMWEIGHTS.

Referenced by C_get_node().

5.18.2.16 SEXP S3get_surrogatesplits (SEXP *node*)

Definition at line 88 of file S3Classes.c.

References S3_SSPLIT.

Referenced by C_get_node(), C_splitsurrogate(), C_surrogates(), and R_surrogates().

5.18.2.17 SEXP S3get_table (SEXP *split*)

Definition at line 192 of file S3Classes.c.

References S3_TABLE.

Referenced by C_Node().

5.18.2.18 SEXP S3get_teststat (SEXP *node*)

Definition at line 64 of file S3Classes.c.

References S3_CRITERION, and S3_STATISTICS.

Referenced by C_Node().

5.18.2.19 int S3get_toleft (SEXP *split*)

Definition at line 170 of file S3Classes.c.

References S3_TOLEFT.

Referenced by C_get_node(), and C_splitsurrogate().

5.18.2.20 int S3get_variableID (SEXP *split*)

Definition at line 154 of file S3Classes.c.

References S3_VARIABLEID.

Referenced by C_get_node(), C_splitnode(), C_splitsurrogate(), and C_surrogates().

5.18.2.21 int S3is_ordered (SEXP *split*)

Definition at line 158 of file S3Classes.c.

References S3_ORDERED.

Referenced by C_get_node(), and C_splitnode().

5.18.2.22 void S3set_nodeID (SEXP *node*, int *nodeID*)

Definition at line 43 of file S3Classes.c.

References S3_NODEID.

Referenced by C_TreeGrow().

5.18.2.23 void S3set_nodeterminal (SEXP *node*)

Definition at line 76 of file S3Classes.c.

References S3_TERMINAL.

5.18.2.24 void S3set_nominal (SEXP *split*)

Definition at line 166 of file S3Classes.c.

References S3_ORDERED.

5.18.2.25 void S3set_ordered (SEXP *split*)

Definition at line 162 of file S3Classes.c.

References S3_ORDERED.

5.18.2.26 void S3set_toleft (SEXP *split*, int *left*)

Definition at line 174 of file S3Classes.c.

References S3_TOLEFT.

5.18.2.27 void S3set_variableID (SEXP *split*, int *variableID*)

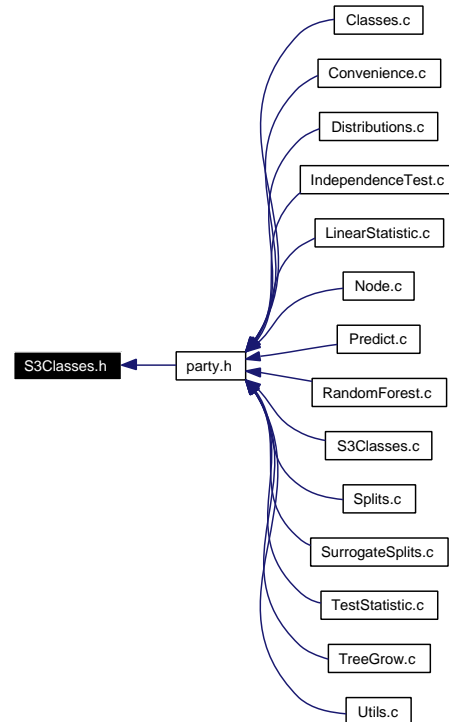
Definition at line 150 of file S3Classes.c.

References S3_VARIABLEID.

Referenced by C_Node().

5.19 S3Classes.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_init_node](#) (SEXP node, int nob, int ninputs, int nsurr, int q)
- void [S3set_nodeID](#) (SEXP node, int nodeID)
- int [S3get_nodeID](#) (SEXP node)
- SEXP [S3get_nodeweights](#) (SEXP node)
- double [S3get_sumweights](#) (SEXP node)
- SEXP [S3get_teststat](#) (SEXP node)
- SEXP [S3get_criterion](#) (SEXP node)
- SEXP [S3get_maxcriterion](#) (SEXP node)
- void [S3set_nodeterminal](#) (SEXP node)
- int [S3get_nodeterminal](#) (SEXP node)
- SEXP [S3get_primarysplit](#) (SEXP node)
- SEXP [S3get_surrogatesplits](#) (SEXP node)
- SEXP [S3get_prediction](#) (SEXP node)
- void [C_init_orderedsplit](#) (SEXP split, int nob)
- void [C_init_nominalsplit](#) (SEXP split, int nlevels, int nob)
- void [S3set_variableID](#) (SEXP split, int variableID)
- int [S3get_variableID](#) (SEXP split)
- int [S3is_ordered](#) (SEXP split)
- void [S3set_ordered](#) (SEXP split)
- void [S3set_nominal](#) (SEXP split)

- SEXP [S3get_splitpoint](#) (SEXP split)
- SEXP [S3get_splitstatistics](#) (SEXP split)
- SEXP [S3get_leftnode](#) (SEXP node)
- SEXP [S3get_rightnode](#) (SEXP node)
- SEXP [S3get_table](#) (SEXP node)
- int [S3get_toleft](#) (SEXP split)
- void [S3set_toleft](#) (SEXP split, int left)

5.19.1 Function Documentation

5.19.1.1 void C_init_node (SEXP *node*, int *nobs*, int *ninputs*, int *nsurr*, int *q*)

Definition at line 11 of file S3Classes.c.

References `CRITERION_LENGTH`, `NODE_LENGTH`, `S3_CRITERION`, `S3_iCRITERION`, `S3_MAXCRITERION`, `S3_NODEID`, `S3_PREDICTION`, `S3_PSPLIT`, `S3_SSPLIT`, `S3_STATISTICS`, `S3_SUMWEIGHTS`, `S3_TERMINAL`, `S3_WEIGHTS`, and `SPLIT_LENGTH`.

Referenced by `C_splitnode()`, `R_Node()`, and `R_TreeGrow()`.

5.19.1.2 void C_init_nominalsplit (SEXP *split*, int *nlevels*, int *nobs*)

Definition at line 128 of file S3Classes.c.

References `S3_ORDERED`, `S3_SPLITPOINT`, `S3_SPLITSTATISTICS`, `S3_TABLE`, `S3_TOLEFT`, `S3_VARIABLEID`, and `SPLIT_LENGTH`.

Referenced by `C_Node()`.

5.19.1.3 void C_init_orderedsplit (SEXP *split*, int *nobs*)

Definition at line 104 of file S3Classes.c.

References `S3_ORDERED`, `S3_SPLITPOINT`, `S3_SPLITSTATISTICS`, `S3_TABLE`, `S3_TOLEFT`, `S3_VARIABLEID`, and `SPLIT_LENGTH`.

Referenced by `C_Node()`.

5.19.1.4 SEXP S3get_criterion (SEXP *node*)

Definition at line 68 of file S3Classes.c.

References `S3_CRITERION`, and `S3_iCRITERION`.

Referenced by `C_Node()`.

5.19.1.5 SEXP S3get_leftnode (SEXP *node*)

Definition at line 96 of file S3Classes.c.

References `S3_LEFT`.

Referenced by `C_get_node()`, `C_get_nodebynum()`, `C_remove_weights()`, `C_splitsurrogate()`, and `C_TreeGrow()`.

5.19.1.6 SEXP S3get_maxcriterion (SEXP *node*)

Definition at line 72 of file S3Classes.c.

References S3_CRITERION, and S3_MAXCRITERION.

Referenced by C_get_node(), and C_Node().

5.19.1.7 int S3get_nodeID (SEXP *node*)

Definition at line 47 of file S3Classes.c.

References S3_NODEID.

Referenced by C_get_nodebynum(), and C_get_nodeID().

5.19.1.8 int S3get_nodeterminal (SEXP *node*)

Definition at line 80 of file S3Classes.c.

References S3_TERMINAL.

Referenced by C_get_node(), C_get_nodebynum(), C_remove_weights(), and C_TreeGrow().

5.19.1.9 SEXP S3get_nodeweights (SEXP *node*)

Definition at line 51 of file S3Classes.c.

References S3_WEIGHTS.

Referenced by C_get_nodeweights(), C_splitnode(), C_splitsurrogate(), C_surrogates(), C_TreeGrow(), and R_TreeGrow().

5.19.1.10 SEXP S3get_prediction (SEXP *node*)

Definition at line 92 of file S3Classes.c.

References S3_PREDICTION.

Referenced by C_get_prediction(), C_getpredictions(), C_Node(), and R_predictRF_weights().

5.19.1.11 SEXP S3get_primarysplit (SEXP *node*)

Definition at line 84 of file S3Classes.c.

References S3_PSPLIT.

Referenced by C_get_node(), C_Node(), C_splitnode(), C_splitsurrogate(), and C_surrogates().

5.19.1.12 SEXP S3get_rightnode (SEXP *node*)

Definition at line 100 of file S3Classes.c.

References S3_RIGHT.

Referenced by C_get_node(), C_get_nodebynum(), C_remove_weights(), C_splitsurrogate(), and C_TreeGrow().

5.19.1.13 SEXP S3get_splitpoint (SEXP *split*)

Definition at line 179 of file S3Classes.c.

References S3_SPLITPOINT.

Referenced by C_get_node(), C_Node(), C_splitnode(), and C_splitsurrogate().

5.19.1.14 SEXP S3get_splitstatistics (SEXP *split*)

Definition at line 183 of file S3Classes.c.

References S3_SPLITSTATISTICS.

Referenced by C_Node().

5.19.1.15 double S3get_sumweights (SEXP *node*)

Definition at line 60 of file S3Classes.c.

References S3_SUMWEIGHTS.

Referenced by C_get_node().

5.19.1.16 SEXP S3get_surrogatesplits (SEXP *node*)

Definition at line 88 of file S3Classes.c.

References S3_SSPLIT.

Referenced by C_get_node(), C_splitsurrogate(), C_surrogates(), and R_surrogates().

5.19.1.17 SEXP S3get_table (SEXP *node*)

Definition at line 192 of file S3Classes.c.

References S3_TABLE.

Referenced by C_Node().

5.19.1.18 SEXP S3get_teststat (SEXP *node*)

Definition at line 64 of file S3Classes.c.

References S3_CRITERION, and S3_STATISTICS.

Referenced by C_Node().

5.19.1.19 int S3get_toleft (SEXP *split*)

Definition at line 170 of file S3Classes.c.

References S3_TOLEFT.

Referenced by C_get_node(), and C_splitsurrogate().

5.19.1.20 int S3get_variableID (SEXP *split*)

Definition at line 154 of file S3Classes.c.

References S3_VARIABLEID.

Referenced by C_get_node(), C_splitnode(), C_splitsurrogate(), and C_surrogates().

5.19.1.21 int S3is_ordered (SEXP *split*)

Definition at line 158 of file S3Classes.c.

References S3_ORDERED.

Referenced by C_get_node(), and C_splitnode().

5.19.1.22 void S3set_nodeID (SEXP *node*, int *nodeID*)

Definition at line 43 of file S3Classes.c.

References S3_NODEID.

Referenced by C_TreeGrow().

5.19.1.23 void S3set_nodeterminal (SEXP *node*)

Definition at line 76 of file S3Classes.c.

References S3_TERMINAL.

5.19.1.24 void S3set_nominal (SEXP *split*)

Definition at line 166 of file S3Classes.c.

References S3_ORDERED.

5.19.1.25 void S3set_ordered (SEXP *split*)

Definition at line 162 of file S3Classes.c.

References S3_ORDERED.

5.19.1.26 void S3set_toleft (SEXP *split*, int *left*)

Definition at line 174 of file S3Classes.c.

References S3_TOLEFT.

5.19.1.27 void S3set_variableID (SEXP *split*, int *variableID*)

Definition at line 150 of file S3Classes.c.

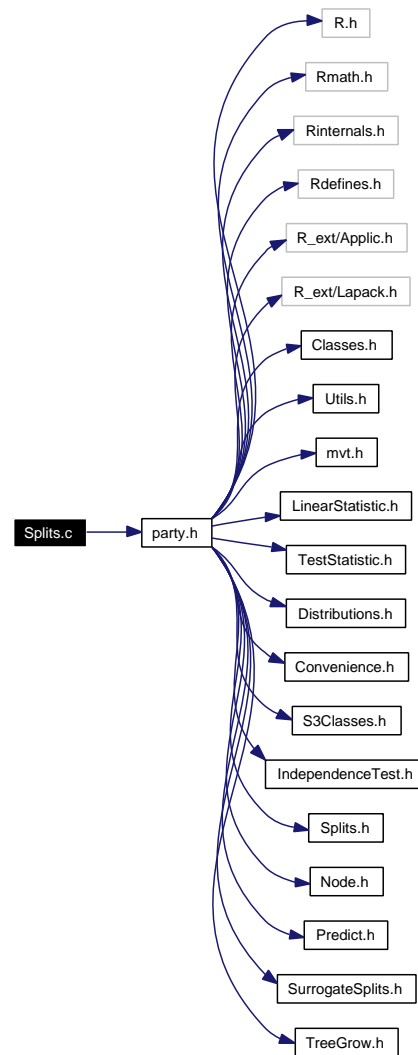
References S3_VARIABLEID.

Referenced by C_Node().

5.20 Splits.c File Reference

```
#include "party.h"
```

Include dependency graph for Splits.c:



Functions

- void [C_split](#) (const double *x, int p, const double *y, int q, const double *weights, int n, const int *orderx, SEXP splitctrl, SEXP linexpcov2sample, SEXP expcovinf, double *cutpoint, double *maxstat, double *statistics)
- SEXP [R_split](#) (SEXP x, SEXP y, SEXP weights, SEXP orderx, SEXP linexpcov2sample, SEXP expcovinf, SEXP splitctrl)
- void [C_splitcategorical](#) (const int *codingx, int p, const double *y, int q, const double *weights, int n, double *standstat, SEXP splitctrl, SEXP linexpcov2sample, SEXP expcovinf, double *cutpoint, int *levelset, double *maxstat, double *statistics)
- SEXP [R_splitcategorical](#) (SEXP x, SEXP codingx, SEXP y, SEXP weights, SEXP linexpcov2sample, SEXP linexpcov, SEXP expcovinf, SEXP splitctrl)

5.20.1 Detailed Description

Binary splits

Author

hothorn

Date

2006-08-25 10:53:10 +0200 (Fri, 25 Aug 2006)

Definition in file [Splits.c](#).

5.20.2 Function Documentation

5.20.2.1 `void C_split (const double * x, int p, const double * y, int q, const double * weights, int n, const int * orderx, SEXP splitctrl, SEXP linexpcov2sample, SEXP expcovinf, double * cutpoint, double * maxstat, double * statistics)`

Search for a cutpoint in a ordered variable *x* maximizing a two-sample statistic w.r.t. (the influence function of) the response variable *y*.

Parameters:

x raw numeric measurements

p dimension of the transformation

y values of the influence function

q dimension of the influence function

weights case weights

n number of observations

orderx the ordering of the transformations, i.e. `R> order(x)`

splitctrl an object of class 'SplitControl'

linexpcov2sample an (uninitialized) object of class 'LinStatExpectCovar' with `p = 1`

expcovinf an initialized object of class 'ExpectCovarInfluence'

cutpoint return value; pointer to a double for the cutpoint in *x*

maxstat return value; pointer to a double for the maximal test statistic

statistics return value; pointer to a n-dim double for the statistics

Definition at line 33 of file `Splits.c`.

References `get_tol()`.

Referenced by `C_Node()`, and `R_split()`.

Here is the call graph for this function:



5.20.2.2 `void C_splitcategorical (const int * codingx, int p, const double * y, int q, const double * weights, int n, double * standstat, SEXP splitctrl, SEXP linexpcov2sample, SEXP expcovinf, double * cutpoint, int * levelset, double * maxstat, double * statistics)`

Search for a cutpoint in a unordered factor *x* maximizing a two-sample statistic w.r.t. (the influence function of) the response variable *y*.

Parameters:

codingx the coding of *x*, i.e. `as.numeric(x)`
p dimension of the transformation
y values of the influence function
q dimension of the influence function
weights case weights
n number of observations
codingx the coding of *x*, i.e. `as.numeric(x)`
standstat the vector of the standardized statistics for *x*, *y*, weights
splitctrl an object of class ‘SplitControl’
linexpcov2sample an (uninitialized) object of class ‘LinStatExpectCovar’ with *p* = 1
expcovinf an initialized object of class ‘ExpectCovarInfluence’
cutpoint return value; pointer to a double for the cutpoint in *x*
levelset return value; pointer to a *p*-dim 0/1 integer
maxstat return value; pointer to a double for the maximal test statistic
statistics return value; pointer to a *n*-dim double for the statistics

Definition at line 217 of file Splits.c.

References `get_tol()`.

Referenced by `C_Node()`, and `R_splitcategorical()`.

Here is the call graph for this function:



5.20.2.3 `SEXP R_split (SEXP x, SEXP y, SEXP weights, SEXP orderx, SEXP linexpcov2sample, SEXP expcovinf, SEXP splitctrl)`

R-interface to `C_split` (does not handle ordered *y*'s)

Parameters:

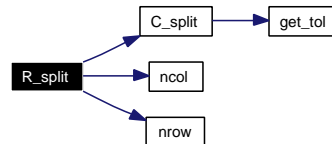
x values of the transformation
y values of the influence function
weights case weights
orderx the ordering of the transformations, i.e. `R> order(x)`
linexpcov2sample an (uninitialized) object of class ‘LinStatExpectCovar’ with *p* = 1
expcovinf an initialized object of class ‘ExpectCovarInfluence’

splitctrl an object of class ‘SplitControl’

Definition at line 175 of file Splits.c.

References C_split(), ncol(), and nrow().

Here is the call graph for this function:



5.20.2.4 SEXP R_splitcategorical (SEXP *x*, SEXP *codingx*, SEXP *y*, SEXP *weights*, SEXP *linexpcov2sample*, SEXP *linexpcov*, SEXP *expcovinf*, SEXP *splitctrl*)

R-interface to C_splitcategorical (does not handle ordered y’s)

Parameters:

x the values of the x-transformation

codingx the coding of x, i.e. as.numeric(x)

y values of the influence function

weights case weights

linexpcov2sample an (uninitialized) object of class ‘LinStatExpectCovar’ with p = 1

linexpcov an initialized object of class ‘LinStatExpectCovar’

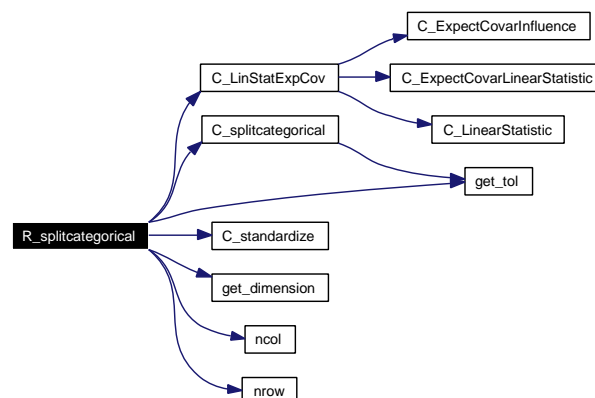
expcovinf an initialized object of class ‘ExpectCovarInfluence’

splitctrl an object of class ‘SplitControl’

Definition at line 307 of file Splits.c.

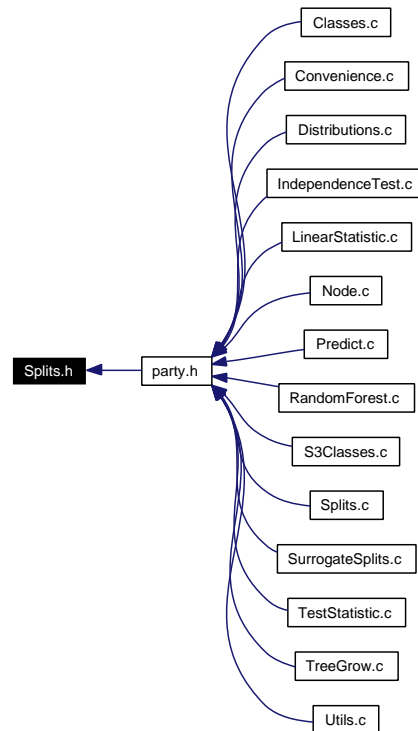
References C_LinStatExpCov(), C_splitcategorical(), C_standardize(), get_dimension(), get_tol(), ncol(), nrow(), PL2_covarianceSym, PL2_expcovinfSym, PL2_expectationSym, and PL2_linearstatisticSym.

Here is the call graph for this function:



5.21 Splits.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_split](#) (const double *x, int p, const double *y, int q, const double *weights, int n, const int *orderx, SEXP splitctrl, SEXP linexpcov2sample, SEXP expcovinf, double *cutpoint, double *maxstat, double *statistics)
- void [C_splitcategorical](#) (const int *codingx, int p, const double *y, int q, const double *weights, int n, double *standstat, SEXP splitctrl, SEXP linexpcov2sample, SEXP expcovinf, double *cutpoint, int *levelset, double *maxstat, double *statistics)

5.21.1 Function Documentation

5.21.1.1 void C_split (const double * x, int p, const double * y, int q, const double * weights, int n, const int * orderx, SEXP splitctrl, SEXP linexpcov2sample, SEXP expcovinf, double * cutpoint, double * maxstat, double * statistics)

Search for a cutpoint in a ordered variable x maximizing a two-sample statistic w.r.t. (the influence function of) the response variable y.

Parameters:

- x* raw numeric measurements
- p* dimension of the transformation
- y* values of the influence function

q dimension of the influence function
weights case weights
n number of observations
orderx the ordering of the transformations, i.e. `R> order(x)`
splitctrl an object of class 'SplitControl'
linexpcov2sample an (uninitialized) object of class 'LinStatExpectCovar' with $p = 1$
expcovinf an initialized object of class 'ExpectCovarInfluence'
cutpoint return value; pointer to a double for the cutpoint in x
maxstat return value; pointer to a double for the maximal test statistic
statistics return value; pointer to a n-dim double for the statistics

Definition at line 33 of file Splits.c.

References `get_tol()`.

Referenced by `C_Node()`, and `R_split()`.

Here is the call graph for this function:



5.21.1.2 `void C_splitcategorical (const int * codingx, int p, const double * y, int q, const double * weights, int n, double * standstat, SEXP splitctrl, SEXP linexpcov2sample, SEXP expcovinf, double * cutpoint, int * levelset, double * maxstat, double * statistics)`

Search for a cutpoint in a unordered factor x maximizing a two-sample statistic w.r.t. (the influence function of) the response variable y .

Parameters:

codingx the coding of x , i.e. `as.numeric(x)`
p dimension of the transformation
y values of the influence function
q dimension of the influence function
weights case weights
n number of observations
codingx the coding of x , i.e. `as.numeric(x)`
standstat the vector of the standardized statistics for x , y , weights
splitctrl an object of class 'SplitControl'
linexpcov2sample an (uninitialized) object of class 'LinStatExpectCovar' with $p = 1$
expcovinf an initialized object of class 'ExpectCovarInfluence'
cutpoint return value; pointer to a double for the cutpoint in x
levelset return value; pointer to a p -dim 0/1 integer
maxstat return value; pointer to a double for the maximal test statistic
statistics return value; pointer to a n -dim double for the statistics

Definition at line 217 of file Splits.c.

References `get_tol()`.

Referenced by `C_Node()`, and `R_splitcategorical()`.

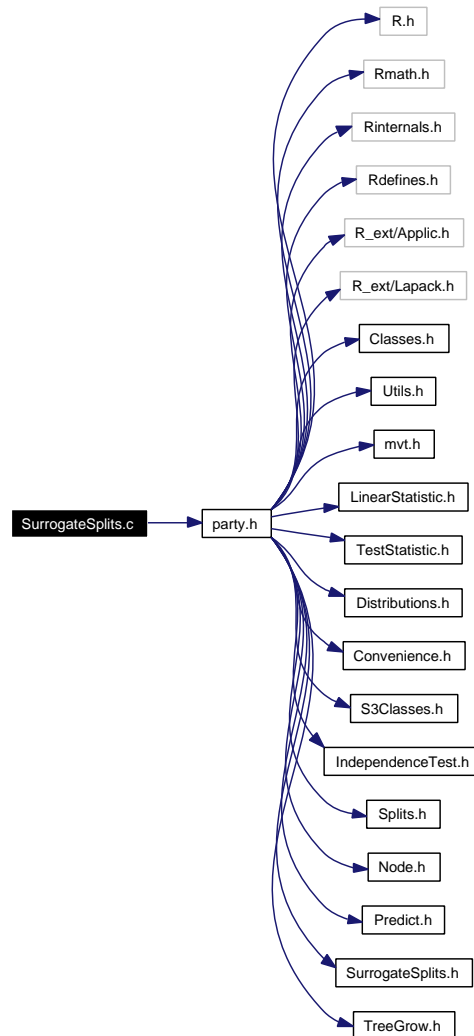
Here is the call graph for this function:



5.22 SurrogateSplits.c File Reference

```
#include "party.h"
```

Include dependency graph for SurrogateSplits.c:



Functions

- void [C_surrogates](#) (SEXP node, SEXP learnsample, SEXP weights, SEXP controls, SEXP fitmem)
- SEXP [R_surrogates](#) (SEXP node, SEXP learnsample, SEXP weights, SEXP controls, SEXP fitmem)
- void [C_splitsurrogate](#) (SEXP node, SEXP learnsample)

5.22.1 Detailed Description

Suggorgate splits

Author

hothorn

Date

2006-09-15 13:27:26 +0200 (Fri, 15 Sep 2006)

Definition in file [SurrogateSplits.c](#).**5.22.2 Function Documentation****5.22.2.1 void C_splitsurrogate (SEXP *node*, SEXP *learnsample*)**

Split with missing values

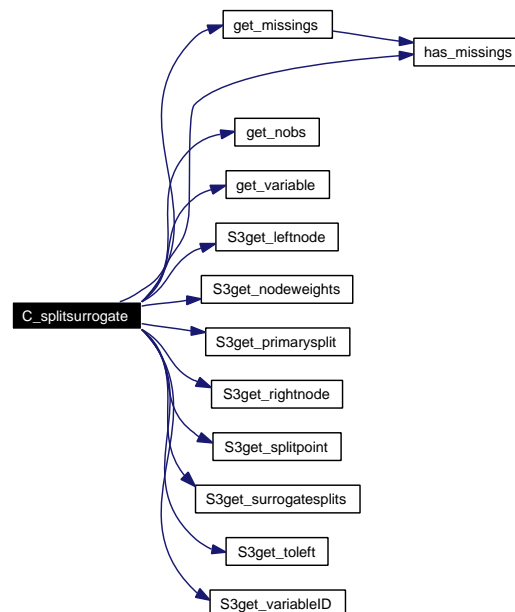
Parameters:*node* the current node with primary and surrogate splits specified*learnsample* learning sample

Definition at line 172 of file SurrogateSplits.c.

References `get_missings()`, `get_nobs()`, `get_variable()`, `has_missings()`, `PL2_inputsSym`, `S3get_leftnode()`, `S3get_nodeweights()`, `S3get_primarysplit()`, `S3get_rightnode()`, `S3get_splitpoint()`, `S3get_surrogatesplits()`, `S3get_toleft()`, and `S3get_variableID()`.

Referenced by `C_TreeGrow()`.

Here is the call graph for this function:

**5.22.2.2 void C_surrogates (SEXP *node*, SEXP *learnsample*, SEXP *weights*, SEXP *controls*, SEXP *ftmem*)**

Search for surrogate splits for bypassing the primary split

Parameters:*node* the current node with primary split specified

learnsample learning sample

weights the weights associated with the current node

controls an object of class ‘TreeControl’

fitmem an object of class ‘TreeFitMemory’

Todo

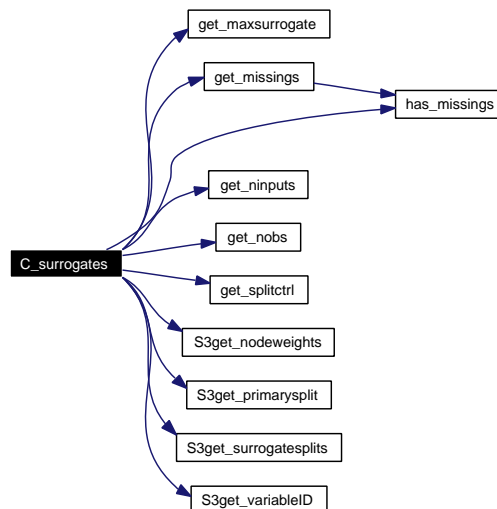
enable nominal surrogate split variables as well

Definition at line 21 of file SurrogateSplits.c.

References `get_maxsurrogate()`, `get_missings()`, `get_ninputs()`, `get_nobs()`, `get_splitctrl()`, `has_missings()`, `PL2_inputsSym`, `S3get_nodeweights()`, `S3get_primarysplit()`, `S3get_surrogatesplits()`, and `S3get_variableID()`.

Referenced by `C_TreeGrow()`, and `R_surrogates()`.

Here is the call graph for this function:



5.22.2.3 SEXP R_surrogates (SEXP node, SEXP learnsample, SEXP weights, SEXP controls, SEXP fitmem)

R-interface to `C_surrogates`

Parameters:

node the current node with primary split specified

learnsample learning sample

weights the weights associated with the current node

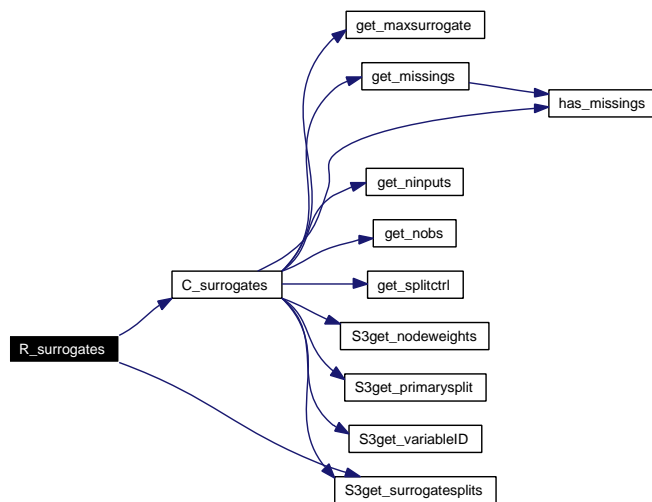
controls an object of class ‘TreeControl’

fitmem an object of class ‘TreeFitMemory’

Definition at line 157 of file SurrogateSplits.c.

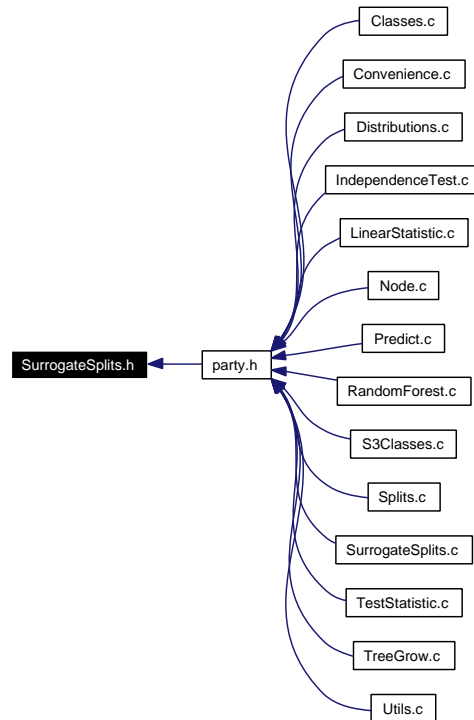
References `C_surrogates()`, and `S3get_surrogatesplits()`.

Here is the call graph for this function:



5.23 SurrogateSplits.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_surrogates](#) (SEXP node, SEXP learnsample, SEXP weights, SEXP controls, SEXP fitmem)
- void [C_splitsurrogate](#) (SEXP node, SEXP learnsample)

5.23.1 Function Documentation

5.23.1.1 void C_splitsurrogate (SEXP *node*, SEXP *learnsample*)

Split with missing values

Parameters:

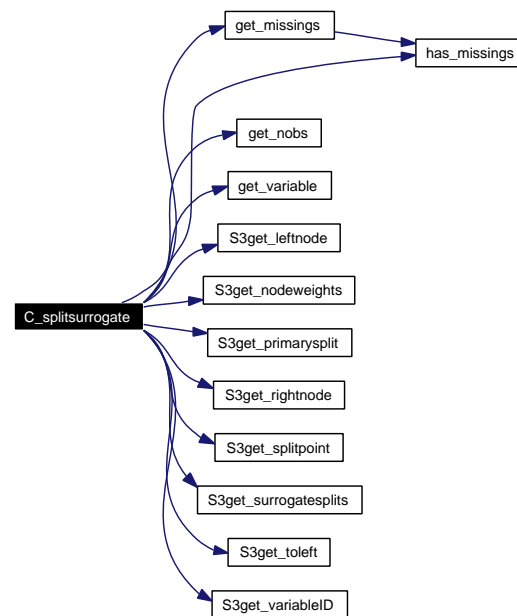
- node* the current node with primary and surrogate splits specified
- learnsample* learning sample

Definition at line 172 of file SurrogateSplits.c.

References [get_missings\(\)](#), [get_nobs\(\)](#), [get_variable\(\)](#), [has_missings\(\)](#), [PL2_inputsSym](#), [S3get_leftnode\(\)](#), [S3get_nodeweights\(\)](#), [S3get_primarysplit\(\)](#), [S3get_rightnode\(\)](#), [S3get_splitpoint\(\)](#), [S3get_surrogatesplits\(\)](#), [S3get_toleft\(\)](#), and [S3get_variableID\(\)](#).

Referenced by [C_TreeGrow\(\)](#).

Here is the call graph for this function:



5.23.1.2 void C_surrogates (SEXP *node*, SEXP *learnsample*, SEXP *weights*, SEXP *controls*, SEXP *fitmem*)

Search for surrogate splits for bypassing the primary split

Parameters:

node the current node with primary split specified

learnsample learning sample

weights the weights associated with the current node

controls an object of class ‘TreeControl’

fitmem an object of class ‘TreeFitMemory’

Todo

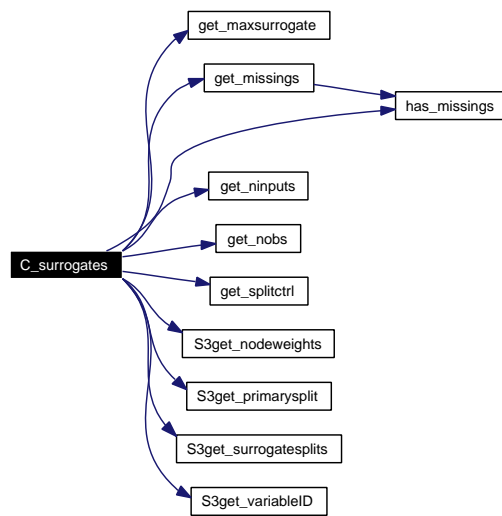
enable nominal surrogate split variables as well

Definition at line 21 of file SurrogateSplits.c.

References `get_maxsurrogate()`, `get_missings()`, `get_ninputs()`, `get_nobs()`, `get_splitctrl()`, `has_missings()`, `PL2_inputsSym`, `S3get_nodeweights()`, `S3get_primarysplit()`, `S3get_surrogatesplits()`, and `S3get_variableID()`.

Referenced by `C_TreeGrow()`, and `R_surrogates()`.

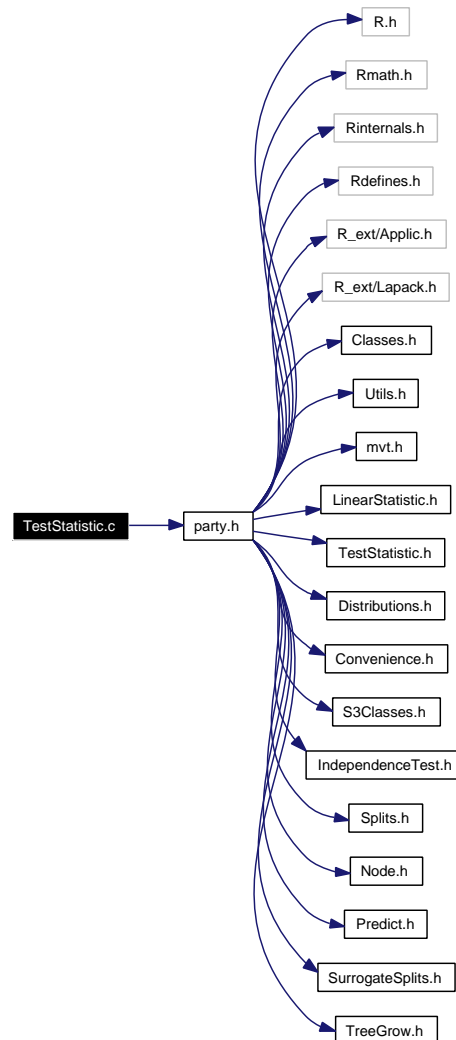
Here is the call graph for this function:



5.24 TestStatistic.c File Reference

```
#include "party.h"
```

Include dependency graph for TestStatistic.c:



Functions

- void [C_standardize](#) (const double *t, const double *mu, const double *Sigma, int pq, double tol, double *ans)
- void [C_absstandardize](#) (const double *t, const double *mu, const double *Sigma, int pq, double tol, double *ans)
- double [C_maxabsTestStatistic](#) (const double *t, const double *mu, const double *Sigma, int pq, double tol)
- SEXP [R_maxabsTestStatistic](#) (SEXP t, SEXP mu, SEXP Sigma, SEXP tol)
- double [C_quadformTestStatistic](#) (const double *t, const double *mu, const double *SigmaPlus, int pq)
- SEXP [R_quadformTestStatistic](#) (SEXP t, SEXP mu, SEXP SigmaPlus)

5.24.1 Detailed Description

Test statistics for conditional inference

Author

hothorn

Date

2005-06-14 11:21:32 +0200 (Tue, 14 Jun 2005)

Definition in file [TestStatistic.c](#).

5.24.2 Function Documentation

5.24.2.1 void C_absstandardize (const double * *t*, const double * *mu*, const double * *Sigma*, int *pq*, double *tol*, double * *ans*)

Absolute values of standardized statistics

Parameters:

t the vector of statistics

mu expectations

Sigma covariance matrix

pq dimension of *t*

tol tolerance for variances

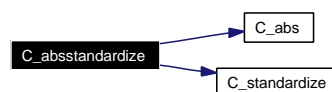
ans return value; a pointer to a REALSXP-vector of length *pq*

Definition at line 49 of file TestStatistic.c.

References C_abs(), and C_standardize().

Referenced by C_maxabsTestStatistic().

Here is the call graph for this function:



5.24.2.2 double C_maxabsTestStatistic (const double * *t*, const double * *mu*, const double * *Sigma*, int *pq*, double *tol*)

Maximum absolute values of standardized statistics

Parameters:

t the vector of statistics

mu expectations

Sigma covariance matrix

pq dimension of *t*

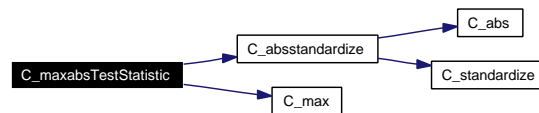
tol tolerance for variances

Definition at line 66 of file TestStatistic.c.

References C_absstandardize(), and C_max().

Referenced by C_TestStatistic(), and R_maxabsTestStatistic().

Here is the call graph for this function:



5.24.2.3 double C_quadformTestStatistic (const double * *t*, const double * *mu*, const double * *SigmaPlus*, int *pq*)

Quadratic form $t(t - \mu) \text{SigmaPlus} (t - \mu)$

Parameters:

t the vector of statistics

mu expectations

SigmaPlus Moore-Penrose inverse

pq dimension of *t*

Definition at line 110 of file TestStatistic.c.

Referenced by C_TestStatistic(), and R_quadformTestStatistic().

5.24.2.4 void C_standardize (const double * *t*, const double * *mu*, const double * *Sigma*, int *pq*, double *tol*, double * *ans*)

Standardizes a statistic *t* of length *pq* with mean *mu* and covariance *Sigma* for variances $> \text{tol}$

Parameters:

t the vector of statistics

mu expectations

Sigma covariance matrix

pq dimension of *t*

tol tolerance for variances

ans return value; a pointer to a REALSXP-vector of length *pq*

Definition at line 23 of file TestStatistic.c.

Referenced by C_absstandardize(), C_Node(), and R_splitcategorical().

5.24.2.5 SEXP R_maxabsTestStatistic (SEXP *t*, SEXP *mu*, SEXP *Sigma*, SEXP *tol*)

R-interface to C_maxabsTestStatistic

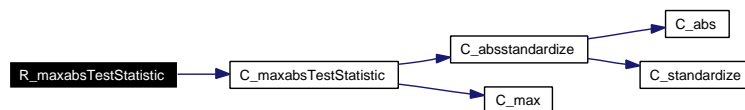
Parameters:

- t* the vector of statistics
- mu* expectations
- Sigma* covariance matrix
- tol* tolerance for variances

Definition at line 87 of file TestStatistic.c.

References C_maxabsTestStatistic().

Here is the call graph for this function:



5.24.2.6 SEXP R_quadformTestStatistic (SEXP *t*, SEXP *mu*, SEXP *SigmaPlus*)

R-interface to C_quadformTestStatistic

Parameters:

- t* the vector of statistics
- mu* expectations
- SigmaPlus* Moore-Penrose inverse

Definition at line 140 of file TestStatistic.c.

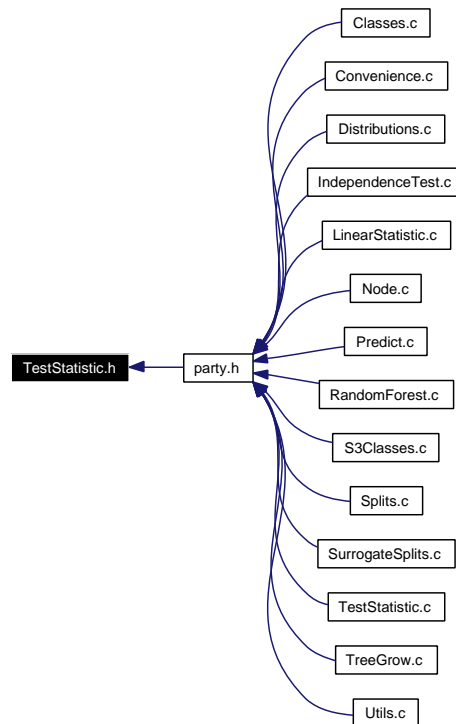
References C_quadformTestStatistic().

Here is the call graph for this function:



5.25 TestStatistic.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_standardize](#) (const double *t, const double *mu, const double *Sigma, int pq, double tol, double *ans)
- double [C_maxabsTestStatistic](#) (const double *t, const double *mu, const double *Sigma, int pq, double tol)
- double [C_quadformTestStatistic](#) (const double *t, const double *mu, const double *SigmaPlus, int pq)

5.25.1 Function Documentation

5.25.1.1 double [C_maxabsTestStatistic](#) (const double * *t*, const double * *mu*, const double * *Sigma*, int *pq*, double *tol*)

Maximum absolute values of standardized statistics

Parameters:

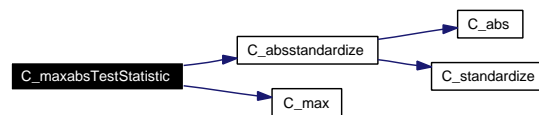
- t* the vector of statistics
- mu* expectations
- Sigma* covariance matrix
- pq* dimension of t
- tol* tolerance for variances

Definition at line 66 of file TestStatistic.c.

References C_absstandardize(), and C_max().

Referenced by C_TestStatistic(), and R_maxabsTestStatistic().

Here is the call graph for this function:



5.25.1.2 double C_quadformTestStatistic (const double * *t*, const double * *mu*, const double * *SigmaPlus*, int *pq*)

Quadratic form $t(t - \mu) \text{SigmaPlus} (t - \mu)$

Parameters:

t the vector of statistics

mu expectations

SigmaPlus Moore-Penrose inverse

pq dimension of *t*

Definition at line 110 of file TestStatistic.c.

Referenced by C_TestStatistic(), and R_quadformTestStatistic().

5.25.1.3 void C_standardize (const double * *t*, const double * *mu*, const double * *Sigma*, int *pq*, double *tol*, double * *ans*)

Standardizes a statistic *t* of length *pq* with mean *mu* and covariance *Sigma* for variances > *tol*

Parameters:

t the vector of statistics

mu expectations

Sigma covariance matrix

pq dimension of *t*

tol tolerance for variances

ans return value; a pointer to a REALSXP-vector of length *pq*

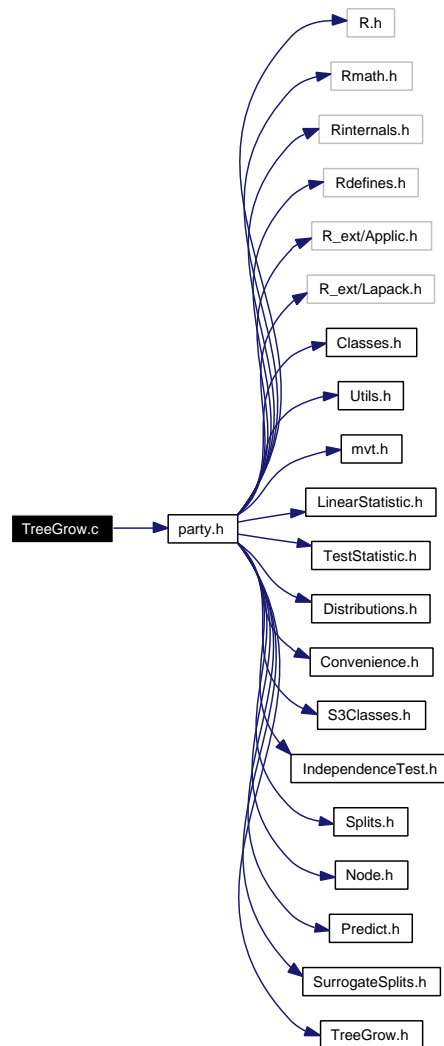
Definition at line 23 of file TestStatistic.c.

Referenced by C_absstandardize(), C_Node(), and R_splitcategorical().

5.26 TreeGrow.c File Reference

```
#include "party.h"
```

Include dependency graph for TreeGrow.c:



Functions

- void [C_TreeGrow](#) (SEXP node, SEXP learnsample, SEXP fitmem, SEXP controls, int *where, int *nodenum, int depth)
- SEXP [R_TreeGrow](#) (SEXP learnsample, SEXP weights, SEXP fitmem, SEXP controls, SEXP where)

5.26.1 Detailed Description

The tree growing recursion

Author

hothorn

Date

2007-07-23 09:44:00 +0200 (Mon, 23 Jul 2007)

Definition in file [TreeGrow.c](#).**5.26.2 Function Documentation****5.26.2.1 void C_TreeGrow (SEXP *node*, SEXP *learnsample*, SEXP *fitmem*, SEXP *controls*, int **where*, int **nodenum*, int *depth*)**

The main tree growing function, handles the recursion.

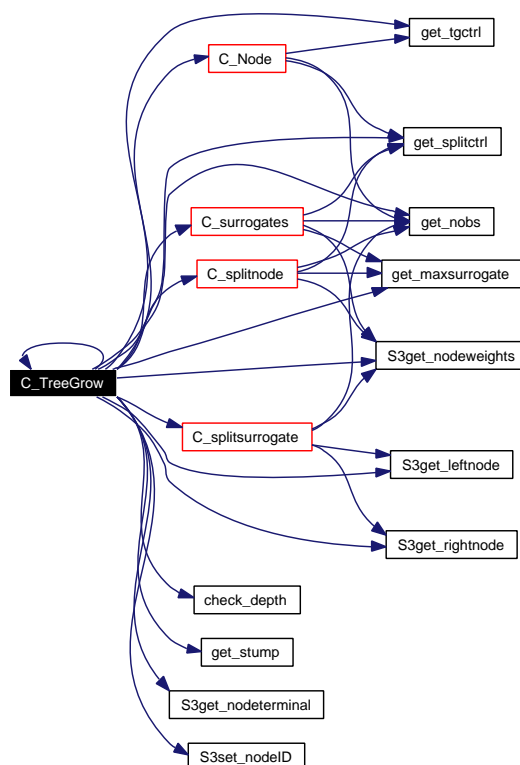
Parameters:*node* a list representing the current node*learnsample* an object of class ‘LearningSample’*fitmem* an object of class ‘TreeFitMemory’*controls* an object of class ‘TreeControl’*where* a pointer to an integer vector of n-elements*nodenum* a pointer to a integer vector of length 1*depth* an integer giving the depth of the current node

Definition at line 23 of file TreeGrow.c.

References C_Node(), C_splitnode(), C_splitsurrogate(), C_surrogates(), C_TreeGrow(), check_depth(), get_maxsurrogate(), get_nobs(), get_splitctrl(), get_stump(), get_tgctrl(), S3get_leftnode(), S3get_nodeterminal(), S3get_nodeweights(), S3get_rightnode(), and S3set_nodeID().

Referenced by C_TreeGrow(), and R_TreeGrow().

Here is the call graph for this function:



5.26.2.2 SEXP R_TreeGrow (SEXP *learnsample*, SEXP *weights*, SEXP *fitmem*, SEXP *controls*, SEXP *where*)

R-interface to `C_TreeGrow`

Parameters:

learnsample an object of class ‘LearningSample’

weights a vector of case weights

fitmem an object of class ‘TreeFitMemory’

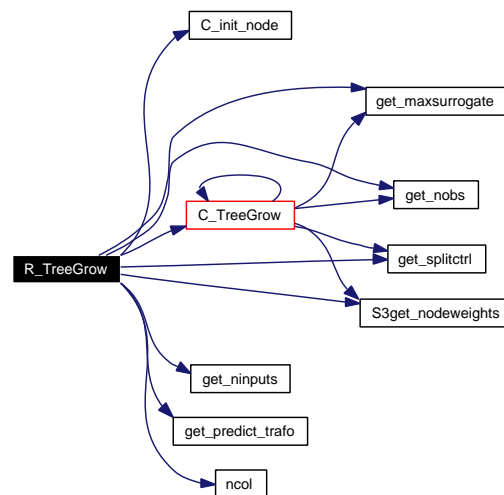
controls an object of class ‘TreeControl’

where a vector of node indices for each observation

Definition at line 81 of file `TreeGrow.c`.

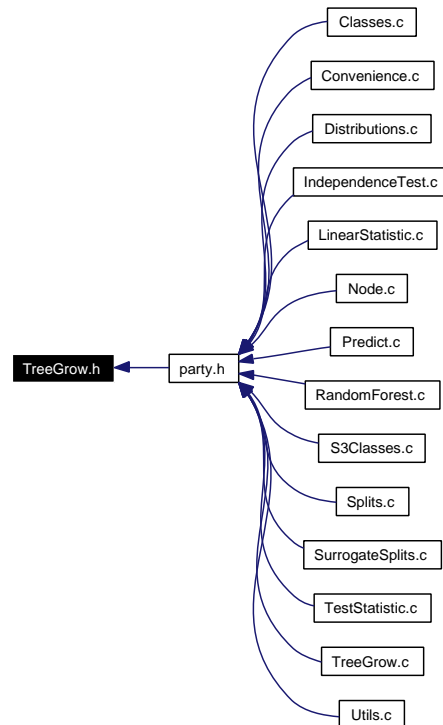
References `C_init_node()`, `C_TreeGrow()`, `get_maxsurrogate()`, `get_ninputs()`, `get_nobs()`, `get_predict_trafo()`, `get_splitctrl()`, `ncol()`, `NODE_LENGTH`, `PL2_responsesSym`, and `S3get_nodeweights()`.

Here is the call graph for this function:



5.27 TreeGrow.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_TreeGrow](#) (SEXP *node*, SEXP *learnsample*, SEXP *fitmem*, SEXP *controls*, int **where*, int **nodenum*, int *depth*)

5.27.1 Function Documentation

5.27.1.1 void [C_TreeGrow](#) (SEXP *node*, SEXP *learnsample*, SEXP *fitmem*, SEXP *controls*, int **where*, int **nodenum*, int *depth*)

The main tree growing function, handles the recursion.

Parameters:

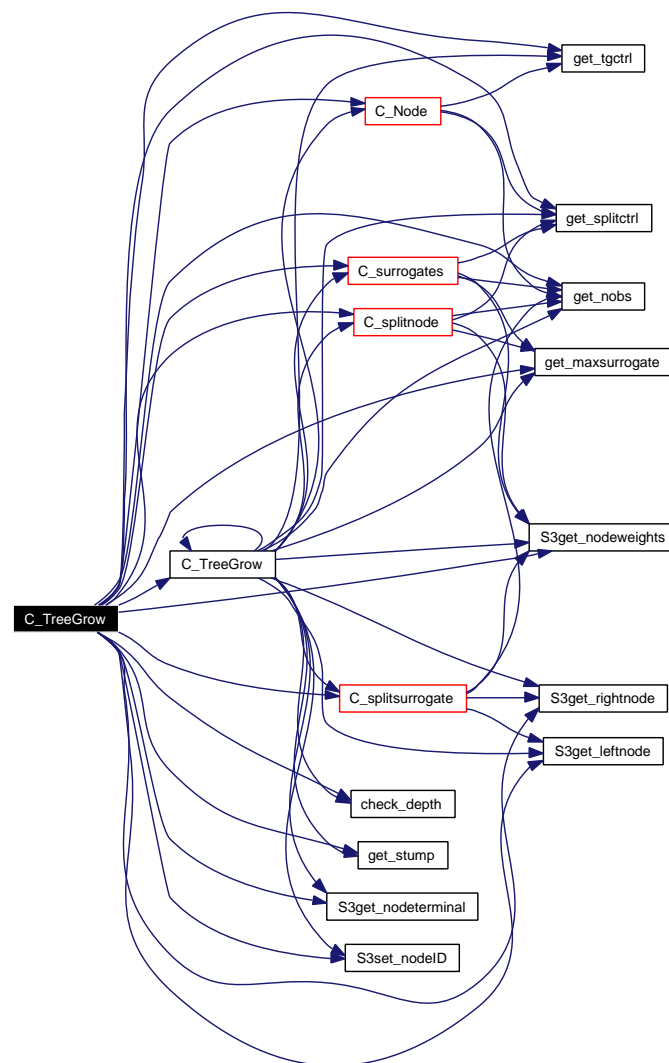
- node* a list representing the current node
- learnsample* an object of class ‘LearningSample’
- fitmem* an object of class ‘TreeFitMemory’
- controls* an object of class ‘TreeControl’
- where* a pointer to an integer vector of n-elements
- nodenum* a pointer to a integer vector of length 1
- depth* an integer giving the depth of the current node

Definition at line 23 of file TreeGrow.c.

References C_Node(), C_splitnode(), C_splitsurrogate(), C_surrogates(), C_TreeGrow(), check_depth(), get_maxsurrogate(), get_nobs(), get_splitctrl(), get_stump(), get_tgctrl(), S3get_leftnode(), S3get_nodeterminal(), S3get_nodeweights(), S3get_rightnode(), and S3set_nodeID().

Referenced by C_TreeGrow(), and R_TreeGrow().

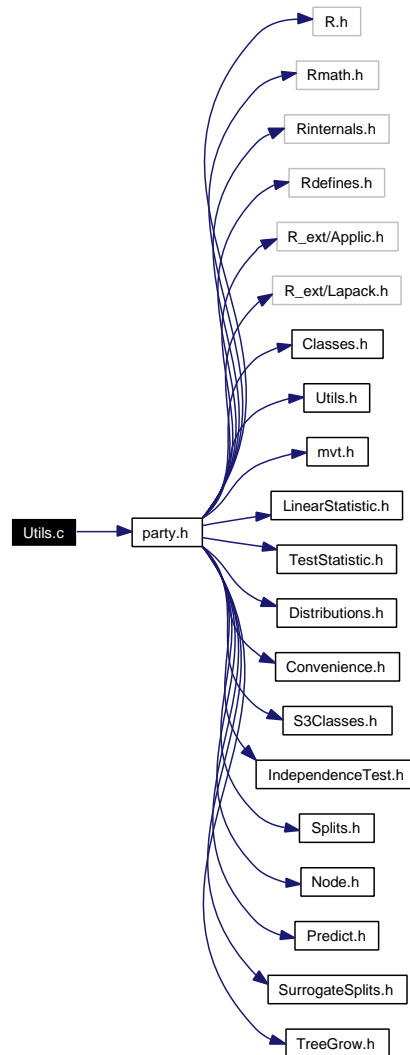
Here is the call graph for this function:



5.28 Utils.c File Reference

```
#include "party.h"
```

Include dependency graph for Utils.c:



Functions

- void [C_kronecker](#) (const double *A, const int m, const int n, const double *B, const int r, const int s, double *ans)
- SEXP [R_kronecker](#) (SEXP A, SEXP B)
- void [CR_La_svd](#) (SEXP jobu, SEXP jobv, SEXP x, SEXP s, SEXP u, SEXP v, SEXP method)
- SEXP [CR_svd](#) (SEXP x, SEXP svdmem)
- void [C_MPinv](#) (SEXP x, double tol, SEXP svdmem, SEXP ans)
- SEXP [R_MPinv](#) (SEXP x, SEXP tol, SEXP svdmem)
- double [C_max](#) (const double *x, const int n)
- SEXP [R_max](#) (SEXP x)
- void [C_abs](#) (double *x, int n)

- `SEXP R_abs` (`SEXP x`)
- `void C_matprod` (`double *x`, `int nrx`, `int ncx`, `double *y`, `int nry`, `int ncy`, `double *z`)
- `SEXP R_matprod` (`SEXP x`, `SEXP y`)
- `void C_matprodT` (`double *x`, `int nrx`, `int ncx`, `double *y`, `int nry`, `int ncy`, `double *z`)
- `SEXP R_matprodT` (`SEXP x`, `SEXP y`)
- `void C_SampleNoReplace` (`int *x`, `int m`, `int k`, `int *ans`)
- `SEXP R_permute` (`SEXP m`)
- `SEXP R_rsubset` (`SEXP m`, `SEXP k`)
- `void C_ProbSampleNoReplace` (`int n`, `double *p`, `int *perm`, `int nans`, `int *ans`)
- `int i_in_set` (`int i`, `int *iset`, `int p`)
- `int C_i_in_set` (`int i`, `SEXP set`)
- `int nrow` (`SEXP x`)
- `int ncol` (`SEXP x`)
- `int C_whichmax` (`double *pvalue`, `double *teststat`, `int ninputs`)
- `SEXP R_whichmax` (`SEXP x`, `SEXP y`)
- `SEXP R_listplus` (`SEXP a`, `SEXP b`, `SEXP which`)
- `SEXP R_modify_response` (`SEXP x`, `SEXP vf`)
- `double F77_SUB() unifrnd` (`void`)
- `void C_SampleSplitting` (`int n`, `double *prob`, `int *weights`, `int k`)
- `void C_remove_weights` (`SEXP subtree`)

5.28.1 Detailed Description

Some commonly needed utility functions.

Auth~~Author~~

hothorn

Date~~Date~~

2007-07-23 12:53:33 +0200 (Mon, 23 Jul 2007)

Definition in file [Utils.c](#).

5.28.2 Function Documentation

5.28.2.1 `void C_abs` (`double * x`, `int n`)

absolute value

Parameters:

x numeric vector

n length(*x*)

Definition at line 315 of file [Utils.c](#).

Referenced by `C_absstandardize()`, and `R_abs()`.

5.28.2.2 int C_i_in_set (int *i*, SEXP *set*)

Definition at line 564 of file Utils.c.

References `i_in_set()`.

Referenced by `C_get_node()`.

Here is the call graph for this function:

**5.28.2.3 void C_kronecker (const double * *A*, const int *m*, const int *n*, const double * *B*, const int *r*, const int *s*, double * *ans*)**

Computes the Kronecker product of two matrices

Parameters:

A matrix

m nrow(*A*)

n ncol(*A*)

B matrix

r nrow(*B*)

s ncol(*B*)

ans return value; a pointer to a REALSXP-vector of length (*mr* x *ns*)

Definition at line 23 of file Utils.c.

Referenced by `R_kronecker()`.

5.28.2.4 void C_matprod (double * *x*, int *nrx*, int *ncx*, double * *y*, int *nry*, int *ncy*, double * *z*)

matrix product $x \% y$

Parameters:

x a matrix

nrx number of rows of *x*

ncx number of cols of *x*

y a matrix

nry number of rows of *y*

ncy number of cols of *y*

z a matrix of dimension *nrx* x *ncy*

Definition at line 353 of file Utils.c.

Referenced by `R_matprod()`.

5.28.2.5 void C_matprodT (double * *x*, int *nrx*, int *ncx*, double * *y*, int *nry*, int *ncy*, double * *z*)

matrix product $x * \% t(y)$

Parameters:

x a matrix
nrx number of rows of *x*
ncx number of cols of *x*
y a matrix
nry number of rows of *y*
ncy number of cols of *y*
z a matrix of dimension *nrx* x *ncy*

Definition at line 405 of file Utils.c.

Referenced by R_matprodT().

5.28.2.6 double C_max (const double * *x*, const int *n*)

the maximum of a double vector

Parameters:

x vector
n its length

Definition at line 278 of file Utils.c.

Referenced by C_maxabsTestStatistic(), C_Node(), and R_max().

5.28.2.7 void C_MPinv (SEXP *x*, double *tol*, SEXP *svdmem*, SEXP *ans*)

Moore-Penrose inverse of a matrix

Parameters:

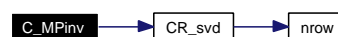
x matrix
tol a tolerance bound
svdmem an object of class 'svd_mem'
ans return value; an object of class 'ExpectCovarMPinv'

Definition at line 185 of file Utils.c.

References CR_svd(), PL2_MPinvSym, PL2_rankSym, PL2_sSym, PL2_uSym, and PL2_vSym.

Referenced by C_LinStatExpCovMPinv(), and R_MPinv().

Here is the call graph for this function:



5.28.2.8 void C_ProbSampleNoReplace (int *n*, double * *p*, int * *perm*, int * *nans*, int * *ans*)

Definition at line 508 of file Utils.c.

Referenced by C_SampleSplitting().

5.28.2.9 void C_remove_weights (SEXP *subtree*)

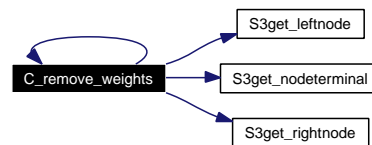
Remove weights vector from each node of a tree (in order to save memory) *param subtree a tree

Definition at line 702 of file Utils.c.

References C_remove_weights(), S3_WEIGHTS, S3get_leftnode(), S3get_nodeterminal(), and S3get_rightnode().

Referenced by C_remove_weights().

Here is the call graph for this function:

**5.28.2.10 void C_SampleNoReplace (int * *x*, int *m*, int *k*, int * *ans*)**

compute a permutation of a (random subset of) 0:(m-1)

Parameters:

x an integer vector of length *m*

m integer

k integer

ans an integer vector of length *k*

Definition at line 453 of file Utils.c.

Referenced by R_permute(), and R_rsubset().

5.28.2.11 void C_SampleSplitting (int *n*, double * *prob*, int * *weights*, int *k*)

Definition at line 679 of file Utils.c.

References C_ProbSampleNoReplace().

Here is the call graph for this function:



5.28.2.12 int C_whichmax (double * *pvalue*, double * *teststat*, int *ninputs*)

Definition at line 583 of file Utils.c.

Referenced by C_Node(), and R_whichmax().

5.28.2.13 void CR_La_svd (SEXP *jobu*, SEXP *jobv*, SEXP *x*, SEXP *s*, SEXP *u*, SEXP *v*, SEXP *method*)

C- and R-interface to La_svd

Parameters:

jobu

jobv

x

s

u

v

method

Definition at line 102 of file Utils.c.

5.28.2.14 SEXP CR_svd (SEXP *x*, SEXP *svdmem*)

C- and R-interface to CR_La_svd

Parameters:

x matrix

svdmem an object of class 'svd_mem'

Definition at line 153 of file Utils.c.

References nrow(), PL2_pSym, PL2_uSym, and PL2_vSym.

Referenced by C_MPinv().

Here is the call graph for this function:



5.28.2.15 int i_in_set (int *i*, int * *iset*, int *p*)

determine if *i* is element of the integer vector set

Parameters:

i an integer

iset a pointer to an integer vector

p length(iset)

Definition at line 549 of file Utils.c.

Referenced by C_i_in_set(), and C_splitnode().

5.28.2.16 int ncol (SEXP *x*)

Definition at line 575 of file Utils.c.

Referenced by C_GlobalTest(), C_IndependenceTest(), C_Node(), C_splitnode(), R_ExpectCovarInfluence(), R_ExpectCovarLinearStatistic(), R_LinearStatistic(), R_matprod(), R_matprodT(), R_MPinv(), R_Node(), R_split(), R_splitcategorical(), and R_TreeGrow().

5.28.2.17 int nrow (SEXP *x*)

Definition at line 571 of file Utils.c.

Referenced by C_IndependenceTest(), CR_svd(), R_ExpectCovarInfluence(), R_ExpectCovarLinearStatistic(), R_LinearStatistic(), R_matprod(), R_matprodT(), R_maxabsConditionalPvalue(), R_MPinv(), R_PermutedLinearStatistic(), R_split(), and R_splitcategorical().

5.28.2.18 SEXP R_abs (SEXP *x*)

R-interface to C_abs

Parameters:

x numeric vector

Definition at line 327 of file Utils.c.

References C_abs().

Here is the call graph for this function:

**5.28.2.19 SEXP R_kronecker (SEXP *A*, SEXP *B*)**

R-interface to C_kronecker

Parameters:

A matrix

B matrix

Definition at line 52 of file Utils.c.

References C_kronecker().

Here is the call graph for this function:

**5.28.2.20 SEXP R_listplus (SEXP *a*, SEXP *b*, SEXP *which*)**

Definition at line 618 of file Utils.c.

5.28.2.21 SEXP R_matprod (SEXP *x*, SEXP *y*)

R-interface to C_matprod

Parameters:

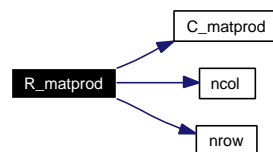
x a matrix

y a matrix

Definition at line 374 of file Utils.c.

References C_matprod(), ncol(), and nrow().

Here is the call graph for this function:



5.28.2.22 SEXP R_matprodT (SEXP *x*, SEXP *y*)

R-interface to C_matprodT

Parameters:

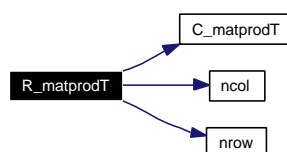
x a matrix

y a matrix

Definition at line 426 of file Utils.c.

References C_matprodT(), ncol(), and nrow().

Here is the call graph for this function:



5.28.2.23 SEXP R_max (SEXP *x*)

R-interface to C_max

Parameters:

x numeric vector

Definition at line 294 of file Utils.c.

References C_max().

Here is the call graph for this function:

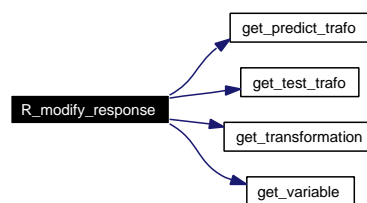


5.28.2.24 SEXP R_modify_response (SEXP *x*, SEXP *vf*)

Definition at line 650 of file Utils.c.

References `get_predict_trafo()`, `get_test_trafo()`, `get_transformation()`, and `get_variable()`.

Here is the call graph for this function:



5.28.2.25 SEXP R_MPinv (SEXP *x*, SEXP *tol*, SEXP *svdmem*)

R-interface to `C_MPinv`

Parameters:

x matrix

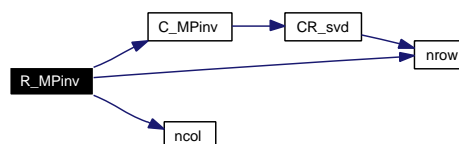
tol a tolerance bound

svdmem an object of class ‘`svd_mem`’

Definition at line 243 of file Utils.c.

References `C_MPinv()`, `ncol()`, `nrow()`, `PL2_MPinvSym`, `PL2_pSym`, and `PL2_rankSym`.

Here is the call graph for this function:



5.28.2.26 SEXP R_permute (SEXP *m*)

R-interface to `C_SampleNoReplace`: the permutation case

Parameters:

m integer

Definition at line 472 of file Utils.c.

References `C_SampleNoReplace()`.

Here is the call graph for this function:



5.28.2.27 SEXP R_rsubset (SEXP *m*, SEXP *k*)

R-interface to C_SampleNoReplace: the subset case

Parameters:

m integer

k integer

Definition at line 492 of file Utils.c.

References C_SampleNoReplace().

Here is the call graph for this function:



5.28.2.28 SEXP R_whichmax (SEXP *x*, SEXP *y*)

Definition at line 608 of file Utils.c.

References C_whichmax().

Here is the call graph for this function:

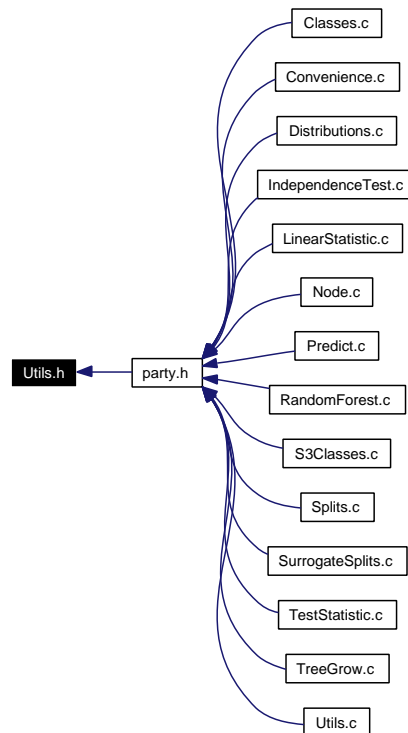


5.28.2.29 double F77_SUB() unifrnd (void)

Definition at line 677 of file Utils.c.

5.29 Utils.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [C_kronecker](#) (const double *A, const int m, const int n, const double *B, const int r, const int s, double *ans)
- SEXP [La_svd](#) (SEXP jobv, SEXP jobv, SEXP x, SEXP s, SEXP u, SEXP v, SEXP method)
- void [C_SampleNoReplace](#) (int *x, int m, int k, int *ans)
- void [C_MPinv](#) (SEXP x, double tol, SEXP svdmem, SEXP ans)
- double [C_max](#) (const double *x, const int n)
- void [C_abs](#) (double *x, int n)
- void [C_matprod](#) (double *x, int nrx, int ncx, double *y, int nry, int ncy, double *z)
- void [C_matprodT](#) (double *x, int nrx, int ncx, double *y, int nry, int ncy, double *z)
- int [nrow](#) (SEXP x)
- int [ncol](#) (SEXP y)
- int [C_whichmax](#) (double *pvalue, double *teststat, int ninputs)
- int [i_in_set](#) (int i, int *iset, int p)
- int [C_i_in_set](#) (int i, SEXP set)
- void [C_SampleSplitting](#) (int n, double *prob, int *weights, int k)
- void [C_remove_weights](#) (SEXP subtree)

5.29.1 Function Documentation

5.29.1.1 void C_abs (double * *x*, int *n*)

absolute value

Parameters:

- x* numeric vector
- n* length(*x*)

Definition at line 315 of file Utils.c.

Referenced by C_absstandardize(), and R_abs().

5.29.1.2 int C_i_in_set (int *i*, SEXP *set*)

Definition at line 564 of file Utils.c.

References i_in_set().

Referenced by C_get_node().

Here is the call graph for this function:



5.29.1.3 void C_kronecker (const double * *A*, const int *m*, const int *n*, const double * *B*, const int *r*, const int *s*, double * *ans*)

Computes the Kronecker product of two matrices

Parameters:

- A* matrix
- m* nrow(*A*)
- n* ncol(*A*)
- B* matrix
- r* nrow(*B*)
- s* ncol(*B*)
- ans* return value; a pointer to a REALSXP-vector of length (mr x ns)

Definition at line 23 of file Utils.c.

Referenced by R_kronecker().

5.29.1.4 void C_matprod (double * *x*, int *nrx*, int *ncx*, double * *y*, int *nry*, int *ncy*, double * *z*)

matrix product $x \% y$

Parameters:

- x* a matrix

nrx number of rows of *x*
ncx number of cols of *x*
y a matrix
nry number of rows of *y*
ncy number of cols of *y*
z a matrix of dimension *nrx* x *ncy*

Definition at line 353 of file Utils.c.

Referenced by R_matprod().

5.29.1.5 void C_matprodT (double * *x*, int *nrx*, int *ncx*, double * *y*, int *nry*, int *ncy*, double * *z*)

matrix product $x * \% t(y)$

Parameters:

x a matrix
nrx number of rows of *x*
ncx number of cols of *x*
y a matrix
nry number of rows of *y*
ncy number of cols of *y*
z a matrix of dimension *nrx* x *ncy*

Definition at line 405 of file Utils.c.

Referenced by R_matprodT().

5.29.1.6 double C_max (const double * *x*, const int *n*)

the maximum of a double vector

Parameters:

x vector
n its length

Definition at line 278 of file Utils.c.

Referenced by C_maxabsTestStatistic(), C_Node(), and R_max().

5.29.1.7 void C_MPinv (SEXP *x*, double *tol*, SEXP *svdmem*, SEXP *ans*)

Moore-Penrose inverse of a matrix

Parameters:

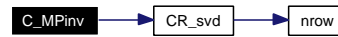
x matrix
tol a tolerance bound
svdmem an object of class 'svd_mem'
ans return value; an object of class 'ExpectCovarMPinv'

Definition at line 185 of file Utils.c.

References CR_svd(), PL2_MPinvSym, PL2_rankSym, PL2_sSym, PL2_uSym, and PL2_vSym.

Referenced by C_LinStatExpCovMPinv(), and R_MPinv().

Here is the call graph for this function:



5.29.1.8 void C_remove_weights (SEXP subtree)

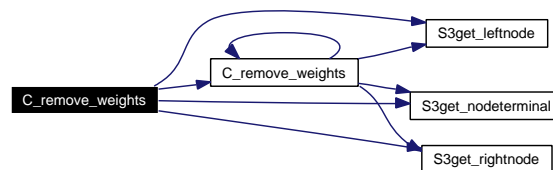
Remove weights vector from each node of a tree (in order to save memory) *param subtree a tree

Definition at line 702 of file Utils.c.

References C_remove_weights(), S3_WEIGHTS, S3get_leftnode(), S3get_nodeterminal(), and S3get_rightnode().

Referenced by C_remove_weights().

Here is the call graph for this function:



5.29.1.9 void C_SampleNoReplace (int * x, int m, int k, int * ans)

compute a permutation of a (random subset of) 0:(m-1)

Parameters:

x an integer vector of length m

m integer

k integer

ans an integer vector of length k

Definition at line 453 of file Utils.c.

Referenced by R_permute(), and R_rsubset().

5.29.1.10 void C_SampleSplitting (int n, double * prob, int * weights, int k)

Definition at line 679 of file Utils.c.

References C_ProbSampleNoReplace().

Here is the call graph for this function:



5.29.1.11 int C_whichmax (double * *pvalue*, double * *teststat*, int *ninputs*)

Definition at line 583 of file Utils.c.

Referenced by C_Node(), and R_whichmax().

5.29.1.12 int i_in_set (int *i*, int * *iset*, int *p*)

determine if *i* is element of the integer vector set

Parameters:

i an integer

iset a pointer to an integer vector

p length(*iset*)

Definition at line 549 of file Utils.c.

Referenced by C_i_in_set(), and C_splitnode().

5.29.1.13 SEXP La_svd (SEXP *jobu*, SEXP *jobv*, SEXP *x*, SEXP *s*, SEXP *u*, SEXP *v*, SEXP *method*)

5.29.1.14 int ncol (SEXP *y*)

Definition at line 575 of file Utils.c.

Referenced by C_GlobalTest(), C_IndependenceTest(), C_Node(), C_splitnode(), R_ExpectCovarInfluence(), R_ExpectCovarLinearStatistic(), R_LinearStatistic(), R_matprod(), R_matprodT(), R_MPinv(), R_Node(), R_split(), R_splitcategorical(), and R_TreeGrow().

5.29.1.15 int nrow (SEXP *x*)

Definition at line 571 of file Utils.c.

Referenced by C_IndependenceTest(), CR_svd(), R_ExpectCovarInfluence(), R_ExpectCovarLinearStatistic(), R_LinearStatistic(), R_matprod(), R_matprodT(), R_maxabsConditionalPvalue(), R_MPinv(), R_PermutedLinearStatistic(), R_split(), and R_splitcategorical().

Chapter 6

party Page Documentation

6.1 Todo List

Member `C_get_node` handle surrogate splits

Member `C_splitnode` outplace the splitting since there are at least 3 functions with nearly identical code

Member `C_surrogates` enable nominal surrogate split variables as well