PriorHessian

Generated by Doxygen 1.8.11

Contents

1 Main Page 1

2 Namespace Index 4

2.1 Namespace List . . . 4

3 Hierarchical Index 4

3.1 Class Hierarchy . . . 4

4 Class Index 7

4.1 Class List . . . 7

5 File Index 9

5.1 File List . . . 9

6 Namespace Documentation 11

6.1 prior_hessian Namespace Reference . . . 12

6.1.1 Typedef Documentation . . . 14

6.1.2 Function Documentation . . . 16

6.2 prior_hessian::constants Namespace Reference . . . 29

6.2.1 Variable Documentation . . . 29

6.3 prior_hessian::CopulaDistImpl Namespace Reference . . . 29

6.4 prior_hessian::detail Namespace Reference . . . 30

6.4.1 Typedef Documentation . . . 30

6.4.2 Function Documentation . . . 30

6.5 prior_hessian::genz Namespace Reference . . . 32

6.5.1 Function Documentation . . . 32

6.6 prior_hessian::genz::fortran Namespace Reference . . . 33

6.6.1 Function Documentation . . . 33

6.7 prior_hessian::helpers Namespace Reference . . . 33

6.7.1 Function Documentation . . . 33

6.8 prior_hessian::mcmc Namespace Reference . . . 34

6.9 prior_hessian::meta Namespace Reference . . . 34

6.9.1 Detailed Description . . . 36

6.9.2 Typedef Documentation . . . 37

6.9.3 Function Documentation . . . 40

6.10 prior_hessian::polylog Namespace Reference . . . 43

6.10.1 Function Documentation . . . 43

7 Class Documentation **45**

7.1 prior_hessian::meta::all_dists_are_bounded< DistTs > Struct Template Reference 45

7.1.1 Detailed Description 46

7.1.2 Member Data Documentation 46

7.2 prior_hessian::AMHCopula< Ndim > Class Template Reference 46

7.2.1 Detailed Description 48

7.2.2 Member Typedef Documentation 48

7.2.3 Constructor & Destructor Documentation 49

7.2.4 Member Function Documentation 49

7.3 prior_hessian::ArchimedeanCopula Class Reference 65

7.3.1 Detailed Description 66

7.4 prior_hessian::BaseDist Class Reference 66

7.4.1 Detailed Description 66

7.5 prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > > Class Template Reference 67

7.5.1 Detailed Description 68

7.5.2 Member Typedef Documentation 69

7.5.3 Constructor & Destructor Documentation 69

7.5.4 Member Function Documentation 69

7.6 prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > > Class Template Reference 78

7.6.1 Detailed Description 79

7.6.2 Member Typedef Documentation 80

7.6.3 Constructor & Destructor Documentation 80

7.6.4 Member Function Documentation 80

7.7 prior_hessian::CompositeDist Class Reference 89

7.7.1 Detailed Description 92

7.7.2 Class Documentation 92

7.7.3 Member Typedef Documentation 93

7.7.4 Constructor & Destructor Documentation . . . 93
7.7.5 Member Function Documentation . . . 95
7.8 prior_hessian::meta::conjunction<... > Struct Template Reference . . . 134
7.8.1 Detailed Description . . . 134
7.9 prior_hessian::meta::conjunction< B1 > Struct Template Reference . . . 135
7.9.1 Detailed Description . . . 135
7.10 prior_hessian::meta::conjunction< B1, Bn... > Struct Template Reference . . . 136
7.10.1 Detailed Description . . . 136
7.11 prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs > Class Template Reference . . . 137
7.11.1 Detailed Description . . . 139
7.11.2 Member Typedef Documentation . . . 139
7.11.3 Constructor & Destructor Documentation . . . 140
7.11.4 Member Function Documentation . . . 141
7.12 prior_hessian::ArchimedeanCopula::D2_GenTerms Struct Reference . . . 159
7.12.1 Detailed Description . . . 160
7.12.2 Member Data Documentation . . . 161
7.13 prior_hessian::ArchimedeanCopula::D2_IGenTerms Struct Reference . . . 161
7.13.1 Detailed Description . . . 162
7.13.2 Member Data Documentation . . . 162
7.14 prior_hessian::ArchimedeanCopula::D2Theta_GenTerms Struct Reference . . . 163
7.14.1 Detailed Description . . . 164
7.14.2 Member Data Documentation . . . 164
7.15 prior_hessian::ArchimedeanCopula::D2Theta_IGenTerms Struct Reference . . . 164
7.15.1 Detailed Description . . . 165
7.15.2 Member Data Documentation . . . 165
7.16 prior_hessian::ArchimedeanCopula::D_GenTerms Struct Reference . . . 166
7.16.1 Detailed Description . . . 166

7.16.2 Member Data Documentation . . . 166
7.17 prior_hessian::ArchimedeanCopula::D_IGenTerms Struct Reference . . . 167
7.17.1 Detailed Description . . . 167
7.17.2 Member Data Documentation . . . 167
7.18 prior_hessian::meta::disjunction<... > Struct Template Reference . . . 168
7.18.1 Detailed Description . . . 169
7.19 prior_hessian::meta::disjunction< B1 > Struct Template Reference . . . 169
7.19.1 Detailed Description . . . 170
7.20 prior_hessian::meta::disjunction< B1, Bn... > Struct Template Reference . . . 170
7.20.1 Detailed Description . . . 171
7.21 prior_hessian::detail::dist_adaptor_traits< Dist > Class Template Reference . . . 171
7.21.1 Detailed Description . . . 171
7.21.2 Member Typedef Documentation . . . 171
7.21.3 Member Data Documentation . . . 172
7.22 prior_hessian::detail::dist_adaptor_traits< CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... > > Struct Template Reference . . . 172
7.22.1 Detailed Description . . . 172
7.22.2 Member Typedef Documentation . . . 172
7.22.3 Member Data Documentation . . . 173
7.23 prior_hessian::detail::dist_adaptor_traits< GammaDist > Struct Template Reference . . . 173
7.23.1 Detailed Description . . . 173
7.23.2 Member Typedef Documentation . . . 173
7.23.3 Member Data Documentation . . . 174
7.24 prior_hessian::detail::dist_adaptor_traits< MultivariateNormalDist< Ndim > > Struct Template Reference 174
7.24.1 Detailed Description . . . 174
7.24.2 Member Typedef Documentation . . . 174
7.24.3 Member Data Documentation . . . 175
7.25 prior_hessian::detail::dist_adaptor_traits< NormalDist > Struct Template Reference . . . 175

7.25.1 Detailed Description . . . 175
7.25.2 Member Typedef Documentation . . . 175
7.25.3 Member Data Documentation . . . 176
7.26 prior_hessian::detail::dist_adaptor_traits< ParetoDist > Struct Template Reference . . . 176
7.26.1 Detailed Description . . . 176
7.26.2 Member Typedef Documentation . . . 176
7.26.3 Member Data Documentation . . . 177
7.27 prior_hessian::detail::dist_adaptor_traits< ScaledSymmetricBetaDist > Struct Template Reference . . . 177
7.27.1 Detailed Description . . . 177
7.27.2 Member Typedef Documentation . . . 177
7.27.3 Member Data Documentation . . . 178
7.28 prior_hessian::detail::dist_adaptor_traits< SymmetricBetaDist > Struct Template Reference . . . 178
7.28.1 Detailed Description . . . 178
7.28.2 Member Typedef Documentation . . . 178
7.28.3 Member Data Documentation . . . 179
7.29 prior_hessian::detail::dist_adaptor_traits< TruncatedGammaDist > Struct Template Reference . . . 179
7.29.1 Detailed Description . . . 179
7.29.2 Member Typedef Documentation . . . 179
7.29.3 Member Data Documentation . . . 180
7.30 prior_hessian::detail::dist_adaptor_traits< TruncatedMultivariateNormalDist< Ndim > > Struct Template Reference . . . 180
7.30.1 Detailed Description . . . 180
7.30.2 Member Typedef Documentation . . . 180
7.30.3 Member Data Documentation . . . 181
7.31 prior_hessian::detail::dist_adaptor_traits< TruncatedNormalDist > Struct Template Reference . . . 181
7.31.1 Detailed Description . . . 181
7.31.2 Member Typedef Documentation . . . 181
7.31.3 Member Data Documentation . . . 182

7.32 prior_hessian::detail::dist_adaptor_traits< TruncatedParetoDist > Struct Template Reference 182
7.32.1 Detailed Description 182
7.32.2 Member Typedef Documentation 182
7.32.3 Member Data Documentation 183
7.33 prior_hessian::ArchimedeanCopula::DTheta_GenTerms Struct Reference 183
7.33.1 Detailed Description 183
7.33.2 Member Data Documentation 183
7.34 prior_hessian::ArchimedeanCopula::DTheta_IGenTerms Struct Reference 184
7.34.1 Detailed Description 184
7.34.2 Member Data Documentation 184
7.35 prior_hessian::eulerian_number< N, M > Struct Template Reference 185
7.35.1 Detailed Description 186
7.36 prior_hessian::eulerian_number< 0, M > Struct Template Reference 186
7.36.1 Detailed Description 187
7.37 prior_hessian::GammaDist Class Reference 187
7.37.1 Detailed Description 189
7.37.2 Member Typedef Documentation 189
7.37.3 Constructor & Destructor Documentation 189
7.37.4 Member Function Documentation 190
7.38 prior_hessian::IndexError Struct Reference 205
7.38.1 Detailed Description 206
7.38.2 Constructor & Destructor Documentation 207
7.38.3 Member Function Documentation 207
7.38.4 Member Data Documentation 207
7.39 prior_hessian::InvalidOperationError Struct Reference 208
7.39.1 Detailed Description 209
7.39.2 Constructor & Destructor Documentation 209
7.39.3 Member Function Documentation 210

7.39.4 Member Data Documentation . . . 210

7.40 prior_hessian::meta::is_copula< T, U > Class Template Reference . . . 210

7.40.1 Detailed Description . . . 210

7.40.2 Member Data Documentation . . . 211

7.41 prior_hessian::meta::is_numeric_template_of< class, typename > Struct Template Reference . . . 211

7.41.1 Detailed Description . . . 212

7.42 prior_hessian::meta::is_numeric_template_of< ClassNumericTemplate, ClassNumericTemplate< Is... > > Struct Template Reference . . . 212

7.42.1 Detailed Description . . . 213

7.43 prior_hessian::meta::is_subclass_of_numeric_template< T, U > Class Template Reference . . . 213

7.43.1 Detailed Description . . . 213

7.43.2 Member Data Documentation . . . 214

7.44 prior_hessian::meta::is_template_of< class, typename > Struct Template Reference . . . 214

7.44.1 Detailed Description . . . 215

7.45 prior_hessian::meta::is_template_of< ClassTemplate, ClassTemplate< Ts... > > Struct Template Reference . . . 215

7.45.1 Detailed Description . . . 216

7.46 prior_hessian::mcmc::MCMCData< Ndim > Class Template Reference . . . 216

7.46.1 Detailed Description . . . 217

7.46.2 Member Typedef Documentation . . . 217

7.46.3 Constructor & Destructor Documentation . . . 217

7.46.4 Member Function Documentation . . . 217

7.46.5 Member Data Documentation . . . 218

7.47 prior_hessian::MultivariateDist Class Reference . . . 218

7.47.1 Detailed Description . . . 219

7.47.2 Constructor & Destructor Documentation . . . 219

7.47.3 Member Function Documentation . . . 219

7.48 prior_hessian::MultivariateNormalDist< Ndim > Class Template Reference . . . 220

7.48.1 Detailed Description . . . 222

7.48.2 Member Typedef Documentation . . . 222
7.48.3 Constructor & Destructor Documentation . . . 223
7.48.4 Member Function Documentation . . . 224
7.49 prior_hessian::NormalDist Class Reference . . . 248
7.49.1 Detailed Description . . . 250
7.49.2 Member Typedef Documentation . . . 250
7.49.3 Constructor & Destructor Documentation . . . 251
7.49.4 Member Function Documentation . . . 252
7.50 prior_hessian::NotImplementedError Struct Reference . . . 272
7.50.1 Detailed Description . . . 273
7.50.2 Constructor & Destructor Documentation . . . 273
7.50.3 Member Function Documentation . . . 274
7.50.4 Member Data Documentation . . . 274
7.51 prior_hessian::NumericalOverflowError Struct Reference . . . 274
7.51.1 Detailed Description . . . 275
7.51.2 Constructor & Destructor Documentation . . . 275
7.51.3 Member Function Documentation . . . 276
7.51.4 Member Data Documentation . . . 276
7.52 prior_hessian::ParameterNameError Struct Reference . . . 276
7.52.1 Detailed Description . . . 277
7.52.2 Constructor & Destructor Documentation . . . 277
7.52.3 Member Function Documentation . . . 278
7.52.4 Member Data Documentation . . . 278
7.53 prior_hessian::ParameterNameUniquenessError Struct Reference . . . 278
7.53.1 Detailed Description . . . 279
7.53.2 Constructor & Destructor Documentation . . . 279
7.53.3 Member Function Documentation . . . 280
7.53.4 Member Data Documentation . . . 280

7.54 prior_hessian::ParameterSizeError Struct Reference . . . 280
7.54.1 Detailed Description . . . 281
7.54.2 Constructor & Destructor Documentation . . . 281
7.54.3 Member Function Documentation . . . 282
7.54.4 Member Data Documentation . . . 282
7.55 prior_hessian::ParameterValueError Struct Reference . . . 282
7.55.1 Detailed Description . . . 283
7.55.2 Constructor & Destructor Documentation . . . 283
7.55.3 Member Function Documentation . . . 284
7.55.4 Member Data Documentation . . . 284
7.56 prior_hessian::ParetoDist Class Reference . . . 284
7.56.1 Detailed Description . . . 286
7.56.2 Member Typedef Documentation . . . 286
7.56.3 Constructor & Destructor Documentation . . . 287
7.56.4 Member Function Documentation . . . 287
7.57 prior_hessian::PriorHessianError Class Reference . . . 319
7.57.1 Detailed Description . . . 320
7.57.2 Constructor & Destructor Documentation . . . 320
7.57.3 Member Function Documentation . . . 321
7.57.4 Member Data Documentation . . . 321
7.58 prior_hessian::RuntimeConvergenceError Struct Reference . . . 321
7.58.1 Detailed Description . . . 322
7.58.2 Constructor & Destructor Documentation . . . 322
7.58.3 Member Function Documentation . . . 323
7.58.4 Member Data Documentation . . . 323
7.59 prior_hessian::RuntimeSamplingError Struct Reference . . . 323
7.59.1 Detailed Description . . . 324
7.59.2 Constructor & Destructor Documentation . . . 324

7.59.3 Member Function Documentation . . . 325
7.59.4 Member Data Documentation . . . 325
7.60 prior_hessian::RuntimeTypeError Struct Reference . . . 325
7.60.1 Detailed Description . . . 326
7.60.2 Constructor & Destructor Documentation . . . 326
7.60.3 Member Function Documentation . . . 327
7.60.4 Member Data Documentation . . . 327
7.61 prior_hessian::ScaledDist< Dist > Class Template Reference . . . 327
7.61.1 Detailed Description . . . 329
7.61.2 Constructor & Destructor Documentation . . . 329
7.61.3 Member Function Documentation . . . 331
7.61.4 Member Data Documentation . . . 343
7.62 prior_hessian::SymmetricBetaDist Class Reference . . . 344
7.62.1 Detailed Description . . . 346
7.62.2 Member Typedef Documentation . . . 346
7.62.3 Constructor & Destructor Documentation . . . 346
7.62.4 Member Function Documentation . . . 347
7.63 prior_hessian::TruncatedDist< Dist > Class Template Reference . . . 361
7.63.1 Detailed Description . . . 363
7.63.2 Constructor & Destructor Documentation . . . 363
7.63.3 Member Function Documentation . . . 366
7.63.4 Member Data Documentation . . . 375
7.64 prior_hessian::TruncatedMultivariateDist< Dist > Class Template Reference . . . 377
7.64.1 Detailed Description . . . 379
7.64.2 Constructor & Destructor Documentation . . . 379
7.64.3 Member Function Documentation . . . 379
7.64.4 Member Data Documentation . . . 381
7.65 prior_hessian::UnivariateDist Class Reference . . . 383
7.65.1 Detailed Description . . . 384
7.65.2 Constructor & Destructor Documentation . . . 384
7.65.3 Member Function Documentation . . . 384
7.66 prior_hessian::UpperTruncatedDist< Dist > Class Template Reference . . . 385
7.66.1 Detailed Description . . . 386
7.66.2 Constructor & Destructor Documentation . . . 386
7.66.3 Member Function Documentation . . . 388

8 File Documentation **395**

8.1 AMHCopula.h File Reference . . . 395

8.1.1 Detailed Description . . . 395

8.2 ArchimedeanCopula.h File Reference . . . 396

8.2.1 Detailed Description . . . 397

8.3 BaseDist.h File Reference . . . 397

8.3.1 Detailed Description . . . 397

8.4 BoundsAdaptedDist.h File Reference . . . 398

8.4.1 Detailed Description . . . 399

8.5 CompositeDist.cpp File Reference . . . 400

8.5.1 Detailed Description . . . 401

8.6 CompositeDist.h File Reference . . . 401

8.6.1 Detailed Description . . . 402

8.6.2 Class Documentation . . . 403

8.7 copula.h File Reference . . . 403

8.7.1 Detailed Description . . . 403

8.8 CopulaDist.h File Reference . . . 404

8.8.1 Detailed Description . . . 405

8.9 EulerianPolynomial.h File Reference . . . 405

8.9.1 Detailed Description . . . 406

8.10 GammaDist.cpp File Reference . . . 406

8.10.1 Detailed Description . . . 407

8.11 GammaDist.h File Reference . . . 407

8.11.1 Detailed Description . . . 408

8.12 Meta.h File Reference . . . 408

8.12.1 Detailed Description . . . 411

8.12.2 Macro Definition Documentation . . . 411

8.13 MultivariateDist.h File Reference . . . 412

8.13.1 Detailed Description . . . 413
8.14 MultivariateNormalDist.h File Reference . . . 413
8.14.1 Detailed Description . . . 414
8.15 mvn_cdf.cpp File Reference . . . 414
8.15.1 Detailed Description . . . 415
8.16 mvn_cdf.h File Reference . . . 415
8.16.1 Detailed Description . . . 417
8.17 NormalDist.cpp File Reference . . . 417
8.17.1 Detailed Description . . . 417
8.18 NormalDist.h File Reference . . . 418
8.18.1 Detailed Description . . . 419
8.19 ParetoDist.cpp File Reference . . . 419
8.19.1 Detailed Description . . . 419
8.20 ParetoDist.h File Reference . . . 420
8.20.1 Detailed Description . . . 421
8.21 PolyLog.h File Reference . . . 421
8.21.1 Detailed Description . . . 422
8.22 PriorHessianError.h File Reference . . . 422
8.22.1 Detailed Description . . . 423
8.22.2 Macro Definition Documentation . . . 424
8.23 README.md File Reference . . . 424
8.24 ScaledDist.h File Reference . . . 424
8.24.1 Detailed Description . . . 425
8.25 ScaledSymmetricBetaDist.h File Reference . . . 425
8.25.1 Detailed Description . . . 426
8.26 SymmetricBetaDist.cpp File Reference . . . 426
8.26.1 Detailed Description . . . 427
8.27 SymmetricBetaDist.h File Reference . . . 427

8.27.1 Detailed Description . . . 428
8.28 TruncatedDist.h File Reference . . . 429
8.28.1 Detailed Description . . . 430
8.29 TruncatedGammaDist.h File Reference . . . 430
8.29.1 Detailed Description . . . 431
8.30 TruncatedMultivariateDist.h File Reference . . . 431
8.30.1 Detailed Description . . . 432
8.31 TruncatedMultivariateNormalDist.h File Reference . . . 433
8.31.1 Detailed Description . . . 434
8.32 TruncatedNormalDist.h File Reference . . . 434
8.32.1 Detailed Description . . . 435
8.33 TruncatedParetoDist.h File Reference . . . 435
8.33.1 Detailed Description . . . 436
8.34 UnivariateDist.cpp File Reference . . . 436
8.34.1 Detailed Description . . . 437
8.35 UnivariateDist.h File Reference . . . 437
8.35.1 Detailed Description . . . 438
8.36 UpperTruncatedDist.h File Reference . . . 438
8.36.1 Detailed Description . . . 439
8.37 util.cpp File Reference . . . 439
8.37.1 Detailed Description . . . 440
8.38 util.h File Reference . . . 441
8.38.1 Detailed Description . . . 442

Index **443**

1 Main Page

Prior Hessian

Library for fast computation of log-likelihoods and derivatives of multivariate priors defined as composites of univariate multivariate distributions especially designed for support of Archimedean copulas.

Documentation

The PriorHessian Doxygen documentation can be build with the `OPT_DOC` CMake option and is also available on online:

- `PriorHessian HTML Manual`
- `PriorHessian PDF Manual`
- `PriorHessian github repository`

Installation

The PriorHessian library uses CMake and is designed to be installed either as a native package, or as a relocatable package in an arbitrary install prefix. The default build script will install to the `_install` directory underneath the repository root.

- **Default release build script**
  - builds to `${BUILD_PATH}` if set or default of: `${CMAKE_SOURCE_DIR}\_build\Release`
  - installs to `${INSTALL_PATH}` if set or default of: `${CMAKE_SOURCE_DIR}\_install`

```
1 $ ./build.sh <cmake-extra-opts>
```

- **Default Debug build script**
  - builds to `${BUILD_PATH}` if set or default of: `${CMAKE_SOURCE_DIR}\_build\Debug`
  - installs to `${INSTALL_PATH}` if set or default of: `${CMAKE_SOURCE_DIR}\_install`

```
1 $ ./build.debug.sh <cmake-extra-opts>
```

- **Manual CMake Build**

```
1 $ cmake -H. -B<build-path> -DCMAKE_INSTALL_PREFEX=<install-path> -DCMAKE_BUILD_TYPE=Release
      <other-cmake-args>
2 $ cd <build-path> && make -j<num-procs> install
```

Dependencies

- *`Armadillo`* - A high-performance array library for C++.
- *BLAS* - A BLAS implemenation: `Netlib BLAS reference` or *`OpenBlas`*
- *LAPACK* - A Lapack implemenation: `Netlib LAPACK reference`

Note the `OPT_BLAS_INT64` CMake option controls whether Armadillo uses BLAS and LAPACK libraries that use 64-bit interger indexing. Matlab uses 64-bit by default, and to link PriorHessian to Matlab MEX libraries, this option must be on. Many linux systems only provide 32-bit integer versions of BLAS and Lapack, and the option can be disabled if Matlab support is not a concern and 64-bit support is difficult to manage on

CMake Options

Standard options

- `BUILD_SHARED_LIBS` - Build shared libraries [Default: On]
- `BUILD_STATIC_LIBS` - Build static libraries [Default: On]
- `BUILD_TESTING` - Build tests [Default: On if `CMAKE_BUILD_TYPE=Debug`]
- `OPT_INSTALL_TESTING` - Install tests. [Default: Off]
- `OPT_DOC` - Build and install documentation (enables `make doc` and `make pdf`) [Default: Off]
- `OPT_EXPORT_BUILD_TREE` - Enable CMake export and `find_package(BacktraceException)` support from the build-tree.

Dependency options

- `OPT_BLAS_INT64` - Enable 64-bit integer BLAS library support [Default: Off]

External Projects

These packages are specialized CMake projects. If they are not currently installed on the development machines we use the `AddExternalDependency.cmake` which will automatically download, configure, build and install to the `CMAKE_INSTALL_PREFIX`, enabling their use through the normal CMake `find_package()` system.

- `BacktraceException` - For exception backtraces when debugging (especially in Matlab).

Motivation

For many likelihood-based methods, they can be extended to Bayesian methods like MAP Estimation and MCMC Posterior sampling, by incorporating a prior. This prior must provide fast methods for computing log-likelihood and it's derivatives over the parameter space. The prior log-likelihood, as well as it's gradient and hessian are then added to the equivalent quantities from the likelihood to create a Bayesian objective for MAP Estimation.

Static Polymorphism

The PriorHessian library is designed using static polymorphism (templates), and as such avoids virtual functions for small-grained tasks, and instead uses templates, which allow many small functions to be inlined. This aggressive inlining by the compiler produces log-likelihood, gradient, and hessian functions that are nearly as fast as hand-coded functions. But our flexible [`CompositeDist`]() class is able to be easily created with any mix of [`UnivariateDist`]() and [`MultivariateDist`]() elements.

In `Mappel` we use this ability to create heterogeneous priors for each Model's parameters, (e.g., [x,y,I,bg,sigma]).

Functionally, the PriorHessian library stores sequences of distributions as `std::tuples`. Using this approach as opposed to the runtime polymorphism of using `std::vector<std::unique_ptr<Base>>` gains several advantages. Most importantly, without the need for virtual functions, the tuple-based approach has the ability to inline the many small computational functions that must be combined for every call to compute the log-likelihood or other computationally important quantities.

2 Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

prior_hessian 12

prior_hessian::constants 29

prior_hessian::CopulaDistImpl 29

prior_hessian::detail 30

prior_hessian::genz 32

prior_hessian::genz::fortran 33

prior_hessian::helpers 33

prior_hessian::mcmc 34

prior_hessian::meta
Class templates to utilize sequencing behavior of std::initializer_list expressions 34

prior_hessian::polylog 43

3 Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

prior_hessian::meta::all_dists_are_bounded< **DistTs** > 45

prior_hessian::ArchimedeanCopula 65

prior_hessian::AMHCopula< **Ndim** > 46
B1

prior_hessian::meta::conjunction< **B1** > 135

prior_hessian::meta::disjunction< **B1** > 169

prior_hessian::BaseDist 66

prior_hessian::MultivariateDist 218

prior_hessian::CopulaDistImpl::CopulaDist< **CopulaTemplate, MarginalDistTs** > 137

prior_hessian::MultivariateNormalDist< **Ndim** > 220

prior_hessian::UnivariateDist 383

prior_hessian::GammaDist 187

prior_hessian::NormalDist 248

prior_hessian::ParetoDist 284

prior_hessian::SymmetricBetaDist 344

prior_hessian::CompositeDist::ComponentDistAdaptor< DistT, Enable > 89

prior_hessian::CompositeDist 89

conditional_t

prior_hessian::meta::conjunction< B1, Bn... > 136

prior_hessian::meta::disjunction< B1, Bn... > 170

prior_hessian::ArchimedeanCopula::D_GenTerms 166

prior_hessian::ArchimedeanCopula::D2_GenTerms 159

prior_hessian::ArchimedeanCopula::D_lGenTerms 167

prior_hessian::ArchimedeanCopula::D2_lGenTerms 161

Dist

prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > > 67

prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > > 78

prior_hessian::ScaledDist< Dist > 327

prior_hessian::TruncatedDist< Dist > 361

prior_hessian::TruncatedMultivariateDist< Dist > 377

prior_hessian::UpperTruncatedDist< Dist > 385

prior_hessian::detail::dist_adaptor_traits< Dist > 171

prior_hessian::detail::dist_adaptor_traits< CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... > > 172

prior_hessian::detail::dist_adaptor_traits< GammaDist > 173

prior_hessian::detail::dist_adaptor_traits< MultivariateNormalDist< Ndim > > 174

prior_hessian::detail::dist_adaptor_traits< NormalDist > 175

prior_hessian::detail::dist_adaptor_traits< ParetoDist > 176

prior_hessian::detail::dist_adaptor_traits< ScaledSymmetricBetaDist > 177

prior_hessian::detail::dist_adaptor_traits< SymmetricBetaDist > 178

prior_hessian::detail::dist_adaptor_traits< TruncatedGammaDist > 179

prior_hessian::detail::dist_adaptor_traits< TruncatedMultivariateNormalDist< Ndim > > 180

prior_hessian::detail::dist_adaptor_traits< TruncatedNormalDist > 181

prior_hessian::detail::dist_adaptor_traits< TruncatedParetoDist > 182

prior_hessian::ArchimedeanCopula::DTheta_GenTerms 183

prior_hessian::ArchimedeanCopula::D2Theta_GenTerms 163

prior_hessian::ArchimedeanCopula::DTheta_IGenTerms 184

prior_hessian::ArchimedeanCopula::D2Theta_IGenTerms 164

std::exception

prior_hessian::PriorHessianError 319

prior_hessian::IndexError 205

prior_hessian::InvalidOperationError 208

prior_hessian::NotImplementedError 272

prior_hessian::NumericalOverflowError 274

prior_hessian::ParameterNameError 276

prior_hessian::ParameterNameUniquenessError 278

prior_hessian::ParameterSizeError 280

prior_hessian::ParameterValueError 282

prior_hessian::RuntimeConvergenceError 321

prior_hessian::RuntimeSamplingError 323

prior_hessian::RuntimeTypeError 325

false_type

prior_hessian::meta::is_numeric_template_of< class, typename > 211

prior_hessian::meta::is_template_of< class, typename > 214

integral_constant

prior_hessian::eulerian_number< N, M > 185

prior_hessian::eulerian_number< 0, M > 186

prior_hessian::meta::is_copula< T, U > 210

prior_hessian::meta::is_subclass_of_numeric_template< T, U > 213

prior_hessian::mcmc::MCMCData< Ndim > 216

prior_hessian::mcmc::MCMCData< Dist::num_dim()> 216
true_type

prior_hessian::meta::conjunction<... > 134

prior_hessian::meta::disjunction<... > 168

prior_hessian::meta::is_numeric_template_of< ClassNumericTemplate, ClassNumericTemplate< Is... > > 212

prior_hessian::meta::is_template_of< ClassTemplate, ClassTemplate< Ts... > > 215

4 Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

prior_hessian::meta::all_dists_are_bounded< DistTs > 45

prior_hessian::AMHCopula< Ndim > 46

prior_hessian::ArchimedeanCopula 65

prior_hessian::BaseDist 66

prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > > 67

prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > > 78

prior_hessian::CompositeDist
A probability distribution made of independent component distributions composing groups of 1 or more variables 89

prior_hessian::meta::conjunction<... > 134

prior_hessian::meta::conjunction< B1 > 135

prior_hessian::meta::conjunction< B1, Bn... > 136

prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs > 137

prior_hessian::ArchimedeanCopula::D2_GenTerms 159

prior_hessian::ArchimedeanCopula::D2_IGenTerms 161

prior_hessian::ArchimedeanCopula::D2Theta_GenTerms 163

prior_hessian::ArchimedeanCopula::D2Theta_IGenTerms 164

prior_hessian::ArchimedeanCopula::D_GenTerms 166

prior_hessian::ArchimedeanCopula::D_lGenTerms 167

prior_hessian::meta::disjunction<... > 168

prior_hessian::meta::disjunction< B1 > 169

prior_hessian::meta::disjunction< B1, Bn... > 170

prior_hessian::detail::dist_adaptor_traits< Dist > 171

prior_hessian::detail::dist_adaptor_traits< CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... > > 172

prior_hessian::detail::dist_adaptor_traits< GammaDist > 173

prior_hessian::detail::dist_adaptor_traits< MultivariateNormalDist< Ndim > > 174

prior_hessian::detail::dist_adaptor_traits< NormalDist > 175

prior_hessian::detail::dist_adaptor_traits< ParetoDist > 176

prior_hessian::detail::dist_adaptor_traits< ScaledSymmetricBetaDist > 177

prior_hessian::detail::dist_adaptor_traits< SymmetricBetaDist > 178

prior_hessian::detail::dist_adaptor_traits< TruncatedGammaDist > 179

prior_hessian::detail::dist_adaptor_traits< TruncatedMultivariateNormalDist< Ndim > > 180

prior_hessian::detail::dist_adaptor_traits< TruncatedNormalDist > 181

prior_hessian::detail::dist_adaptor_traits< TruncatedParetoDist > 182

prior_hessian::ArchimedeanCopula::DTheta_GenTerms 183

prior_hessian::ArchimedeanCopula::DTheta_lGenTerms 184

prior_hessian::eulerian_number< N, M > 185

prior_hessian::eulerian_number< 0, M > 186

prior_hessian::GammaDist
Gamma distribution 187

prior_hessian::IndexError
Indicates a index access was out of bounds 205

prior_hessian::InvalidOperationError 208

prior_hessian::meta::is_copula< T, U > 210

prior_hessian::meta::is_numeric_template_of< class, typename > 211

prior_hessian::meta::is_numeric_template_of< ClassNumericTemplate, ClassNumericTemplate< Is... > > 212

prior_hessian::meta::is_subclass_of_numeric_template< T, U > 213

prior_hessian::meta::is_template_of< class, typename > 214

prior_hessian::meta::is_template_of< ClassTemplate, ClassTemplate< Ts... > > 215

prior_hessian::mcmc::MCMCData< Ndim > 216

prior_hessian::MultivariateDist 218

prior_hessian::MultivariateNormalDist< Ndim >
Multivariate Normal distribution 220

prior_hessian::NormalDist
Normal distribution with truncation 248

prior_hessian::NotImplementedError 272

prior_hessian::NumericalOverflowError 274

prior_hessian::ParameterNameError 276

prior_hessian::ParameterNameUniquenessError 278

prior_hessian::ParameterSizeError 280

prior_hessian::ParameterValueError 282

prior_hessian::ParetoDist
Pareto dist with infinite upper bound 284

prior_hessian::PriorHessianError 319

prior_hessian::RuntimeConvergenceError 321

prior_hessian::RuntimeSamplingError 323

prior_hessian::RuntimeTypeError 325

prior_hessian::ScaledDist< Dist > 327

prior_hessian::SymmetricBetaDist
Single parameter beta distribution where $\alpha = \beta$, leading to symmetric bounded distribution 344

prior_hessian::TruncatedDist< Dist > 361

prior_hessian::TruncatedMultivariateDist< Dist > 377

prior_hessian::UnivariateDist 383

prior_hessian::UpperTruncatedDist< Dist > 385

5 File Index

5.1 File List

Here is a list of all files with brief descriptions:

AMHCopula.h
Ali-Mikhail-Haq Archimedean Copula 395

ArchimedeanCopula.h
CopulaDist base class 396

BaseDist.h
The Base class for UnivariateDist and MultivariateDist 397

BoundsAdaptedDist.h
Functions and type-traits to enable easy wrapping of distributions in bounds-modifiable adapters 398

CompositeDist.cpp
CompositeDist and associated classes and nested classes 400

CompositeDist.h
CompositeDist class declaration and inline and templated member function definitions 401

copula.h
CopulaDist base class 403

CopulaDist.h
CopulaDist base class 404

EulerianPolynomial.h
EulerianPolynomial computation 405

GammaDist.cpp
GammaDist class definition 406

GammaDist.h
GammaDist class declaration and templated methods 407

Meta.h
Enables the use of variadic templates in interesting ways 408

MultivariateDist.h
MultivariateDist base class 412

MultivariateNormalDist.h
MultivariateNormalDist base class 413

mvn_cdf.cpp 414

mvn_cdf.h
Numerical computation of multivariate normal cdfs in 2,3 and higher dims 415

NormalDist.cpp
NormalDist class definition 417

NormalDist.h
NormalDist class declaration and templated methods 418

ParetoDist.cpp
ParetoDist class definition 419

ParetoDist.h
ParetoDist class declaration and templated methods 420

PolyLog.h
Poly log computation for negative integer valued parameters 421

PriorHessianError.h
The Exception classes for the PriorHessian library 422

ScaledDist.h
ScaledDist class declaration and templated methods 424

ScaledSymmetricBetaDist.h
ScaledSymmetricBetaDist class declaration and templated methods 425

SymmetricBetaDist.cpp
SymmetricBetaDist class definition 426

SymmetricBetaDist.h
SymmetricBetaDist class declaration and templated methods 427

TruncatedDist.h
TruncatedDist declaration and templated methods definitions 429

TruncatedGammaDist.h
TruncatedGammaDist class declaration and templated methods 430

TruncatedMultivariateDist.h
TruncatedMultivariateDist declaration and templated methods definitions 431

TruncatedMultivariateNormalDist.h
TruncatedMultivariateNormalDist class declaration 433

TruncatedNormalDist.h
TruncatedNormalDist class declaration 434

TruncatedParetoDist.h
TruncatedParetoDist class declaration and templated methods 435

UnivariateDist.cpp
UnivariateDist base class method definition 436

UnivariateDist.h
UnivariateDist base class 437

UpperTruncatedDist.h
UpperTruncatedDist declaration and templated methods definitions 438

util.cpp
Utilities 439

util.h
Utilities and namespace globals 441

6 Namespace Documentation

6.1 prior_hessian Namespace Reference

Namespaces

- constants
- CopulaDistImpl
- detail
- genz
- helpers
- mcmc
- meta

  Class templates to utilize sequencing behavior of std::initializer_list expressions.
- polylog

Classes

- class AMHCopula
- class ArchimedeanCopula
- class BaseDist
- class CompositeDist

  A probability distribution made of independent component distributions composing groups of 1 or more variables.
- struct eulerian_number
- struct eulerian_number< 0, M >
- class GammaDist

  Gamma distribution.
- struct IndexError

  Indicates a index access was out of bounds.
- struct InvalidOperationError
- class MultivariateDist
- class MultivariateNormalDist

  Multivariate Normal distribution.
- class NormalDist

  Normal distribution with truncation.
- struct NotImplementedError
- struct NumericalOverflowError
- struct ParameterNameError
- struct ParameterNameUniquenessError
- struct ParameterSizeError
- struct ParameterValueError
- class ParetoDist

  Pareto dist with infinite upper bound.
- class PriorHessianError
- struct RuntimeConvergenceError
- struct RuntimeSamplingError
- struct RuntimeTypeError
- class ScaledDist
- class SymmetricBetaDist

  Single parameter beta distribution where $\alpha = \beta$, leading to symmetric bounded distribution.
- class TruncatedDist
- class TruncatedMultivariateDist
- class UnivariateDist
- class UpperTruncatedDist

Typedefs

- template<class DistT >
  using BoundsAdaptedDistT = typename detail::dist_adaptor_traits< std::decay_t< DistT >>::bounds_adapted↩
  _dist
- template<template< int > class CopulaTemplate, class... MarginalDistTs>
  using CopulaDist = CopulaDistImpl::CopulaDist< CopulaTemplate, BoundsAdaptedDistT< MarginalDistTs >...
  >
- using ScaledSymmetricBetaDist = ScaledDist< SymmetricBetaDist >
- using TruncatedGammaDist = TruncatedDist< GammaDist >
- template<IdxT Ndim>
  using TruncatedMultivariateNormalDist = TruncatedMultivariateDist< MultivariateNormalDist< Ndim >>
- using TruncatedNormalDist = TruncatedDist< NormalDist >
- using TruncatedParetoDist = UpperTruncatedDist< ParetoDist >
- using IdxT = arma::uword
- using UVecT = arma::Col< IdxT >
- using VecT = arma::Col< double >
- using MatT = arma::Mat< double >
- using StringVecT = std::vector< std::string >
- using TypeInfoVecT = std::vector< std::type_index >

Functions

- std::ostream & operator<< (std::ostream &out, const CompositeDist &comp_dist)
- double unit_normal_cdf (double t)
- double unit_normal_icdf (double u)
- double bounded (double x)
- double owen_t_integral (double h, double a, double gh)
- double owen_b_integral (double h, double k, double r)
- double donnelly_bvn_integral (double ah, double ak, double r)
- double donnelly_bvn_integral_orig (double ah, double ak, double r)
- template<class Dist , typename = meta::EnableIfIsNotTupleT<Dist>>
  std::enable_if_t< detail::DistTraitsT< Dist >::adaptable_bounds, Dist > make_adapted_bounded_dist (Dist
  &&dist)
- template<class Dist , typename = meta::EnableIfIsNotTupleT<Dist>>
  std::enable_if_t< !detail::DistTraitsT< Dist >::adaptable_bounds, BoundsAdaptedDistT< Dist > > make_↩
  adapted_bounded_dist (Dist &&dist)
- template<class Dist , class Vec , typename = meta::EnableIfIsNotTupleT<Dist>>
  std::enable_if_t< detail::DistTraitsT< Dist >::adaptable_bounds, Dist > make_adapted_bounded_dist (Dist
  &&dist, Vec &&lbound, Vec &&ubound)
- template<class Dist , class Vec , typename = meta::EnableIfIsNotTupleT<Dist>>
  std::enable_if_t<!detail::DistTraitsT< Dist >::adaptable_bounds, BoundsAdaptedDistT< Dist > > make_↩
  adapted_bounded_dist (Dist &&dist, Vec &&lbound, Vec &&ubound)
- template<class... Ts>
  std::tuple< BoundsAdaptedDistT< Ts >... > make_adapted_bounded_dist_tuple (Ts &&...ts)
- template<class... Ts>
  std::tuple< BoundsAdaptedDistT< Ts >... > make_adapted_bounded_dist_tuple (std::tuple< Ts... > &&dists)
- template<class... Ts>
  std::tuple< BoundsAdaptedDistT< Ts >... > make_adapted_bounded_dist_tuple (const std::tuple< Ts... >
  &dists)

- template<template< int > class CopulaTemplate, class... MarginalDistTs>
  CopulaDist< CopulaTemplate, MarginalDistTs... > make_copula_dist (CopulaTemplate< sizeof...(Marginal↩
  DistTs)> &&copula, MarginalDistTs &&...dists)
- template<long N>
  VecT eulerian_polynomial ()
- double owen_t_integral (double h, double a)
- template<class Vec , class Mat >
  double donnelly_bvn_cdf (const Vec &b, const Mat &sigma)
- template<class Vec , class Mat >
  double owen_bvn_cdf (const Vec &b, const Mat &sigma)
- template<class Vec , class Mat >
  double mc_mvn_integral (const Vec &a, const Vec &b, const Mat &U, double &error, int &niter)
- template<class Vec , class Mat >
  double mc_mvn_cdf_core (const Vec &b, const Mat &U, double &error, int &niter)
- template<class Vec , class Mat >
  double mc_mvn_cdf (const Vec &b, const Mat &S, double &error)
- ScaledSymmetricBetaDist make_scaled_symmetric_beta_dist (double beta, std::pair< double, double > bounds)
- TruncatedGammaDist make_bounded_gamma_dist (double scale, double shape, std::pair< double, double > bounds)
- template<IdxT Ndim, class Vec , class Mat , class Vec2 >
  TruncatedMultivariateNormalDist< Ndim > make_bounded_multivariate_normal_dist (Vec &&mu, Mat &&sigma, Vec2 &&lbound, Vec2 &&ubound)
- TruncatedNormalDist make_bounded_normal_dist (double mu, double sigma, std::pair< double, double > bounds)
- TruncatedParetoDist make_bounded_pareto_dist (double alpha, std::pair< double, double > bounds)
- template<class T >
  T square (T t)

6.1.1 Typedef Documentation

6.1.1.1 template<class DistT > using prior_hessian::BoundsAdaptedDistT = typedef typename detail::dist_adaptor_traits<std::decay_t<DistT>>::bounds_adapted_dist

The bounds-adapted distribution type for a given distribution type DistT This is the adapted version of the class, i.e., the class that allows truncation or scaling so that the lower and upper bounds are settable.

Definition at line 59 of file BoundsAdaptedDist.h.

6.1.1.2 template<template< int > class CopulaTemplate, class... MarginalDistTs> using prior_hessian::CopulaDist = typedef CopulaDistImpl::CopulaDist<CopulaTemplate, BoundsAdaptedDistT<MarginalDistTs>...>

Definition at line 182 of file CopulaDist.h.

6.1.1.3 using prior_hessian::IdxT = typedef arma::uword

Definition at line 28 of file util.h.

6.1.1.4 using prior_hessian::MatT = typedef arma::Mat<double>

Definition at line 31 of file util.h.

6.1.1.5 using prior_hessian::ScaledSymmetricBetaDist = typedef ScaledDist<SymmetricBetaDist>

Definition at line 15 of file ScaledSymmetricBetaDist.h.

6.1.1.6 using prior_hessian::StringVecT = typedef std::vector<std::string>

Definition at line 32 of file util.h.

6.1.1.7 using prior_hessian::TruncatedGammaDist = typedef TruncatedDist<GammaDist>

Definition at line 15 of file TruncatedGammaDist.h.

6.1.1.8 template<IdxT Ndim> using prior_hessian::TruncatedMultivariateNormalDist = typedef TruncatedMultivariateDist<MultivariateNormalDist<Ndim>>

Definition at line 18 of file TruncatedMultivariateNormalDist.h.

6.1.1.9 using prior_hessian::TruncatedNormalDist = typedef TruncatedDist<NormalDist>

Definition at line 17 of file TruncatedNormalDist.h.

6.1.1.10 using prior_hessian::TruncatedParetoDist = typedef UpperTruncatedDist<ParetoDist>

Definition at line 15 of file TruncatedParetoDist.h.

6.1.1.11 using prior_hessian::TypeInfoVecT = typedef std::vector<std::type_index>

Definition at line 33 of file util.h.

6.1.1.12 using prior_hessian::UVecT = typedef arma::Col<IdxT>

Definition at line 29 of file util.h.

6.1.1.13 using prior_hessian::VecT = typedef arma::Col<double>

Definition at line 30 of file util.h.

6.1.2 Function Documentation

6.1.2.1 double prior_hessian::bounded (double *x*)

Definition at line 42 of file mvn_cdf.cpp.

Referenced by donnelly_bvn_integral().

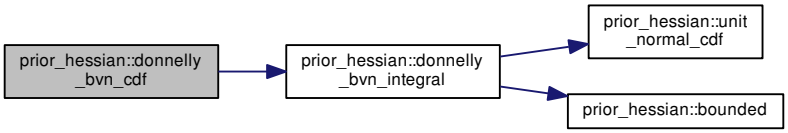
Here is the caller graph for this function:

6.1.2.2 template<class Vec , class Mat > double prior_hessian::donnelly_bvn_cdf (const Vec & *b,* const Mat & *sigma*)

Definition at line 55 of file mvn_cdf.h.

References donnelly_bvn_integral().

Here is the call graph for this function:



6.1.2.3 double prior_hessian::donnelly_bvn_integral (double *ah,* double *ak,* double *r*)

compute the bivariate normal cdf integral computes the probability for two normal variates X and Y whose correlation is R, that AH <= X and AK <= Y.

Adapted to modern C++ with efficiency improvements by: Mark Olah (mjo@cs.unm DOT edu) 10/2018

Reference: Thomas Donnelly, Algorithm 462: Bivariate Normal Distribution, Communications of the ACM, October 1973, Volume 16, Number 10, page 638.

compute the upper-right tail of the bivariate normal distribution computes the probability for two normal variates X and Y whose correlation is R, that AH <= X and AK <= Y.

Adapted to modern C++ with efficiency improvements by: Mark Olah (mjo@cs.unm DOT edu) 10/2018

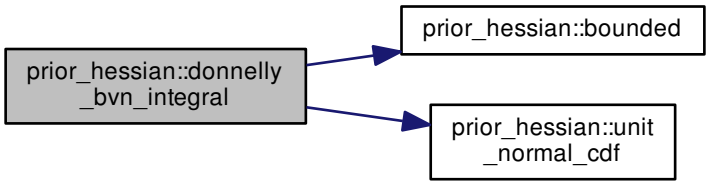
Reference: Thomas Donnelly, Algorithm 462: Bivariate Normal Distribution, Communications of the ACM, October 1973, Volume 16, Number 10, page 638.

Definition at line 158 of file mvn_cdf.cpp.

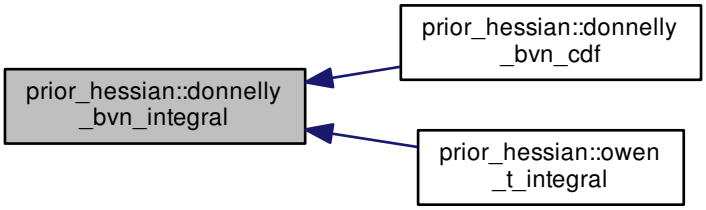
References bounded(), and unit_normal_cdf().

Referenced by donnelly_bvn_cdf(), and owen_t_integral().

Here is the call graph for this function:



Here is the caller graph for this function:



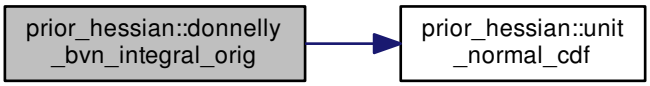
6.1.2.4 double prior_hessian::donnelly_bvn_integral_orig (double *ah*, double *ak*, double *r*)

Definition at line 268 of file mvn_cdf.cpp.

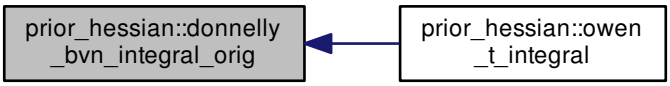
References unit_normal_cdf().

Referenced by owen_t_integral().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.2.5 template<long N> VecT prior_hessian::eulerian_polynomial ( )

Definition at line 33 of file EulerianPolynomial.h.

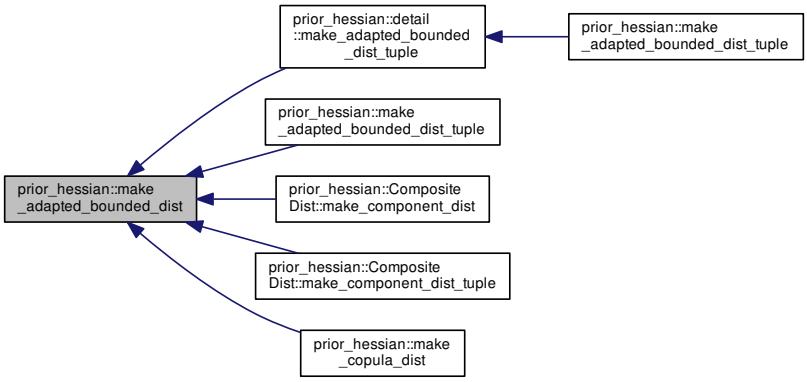
6.1.2.6 template<class Dist , typename = meta::EnableIfIsNotTupleT<Dist>> std::enable_if_t< detail::DistTraitsT<Dist>::adaptable_bounds, Dist> prior_hessian::make_adapted_bounded_dist (Dist && *dist*)

make_adapted_bounded_dist() [4-forms] If the given distribution is not bounded then the appropriate bounding distribution is wrapped around it. We detect the boundedness of the distribution using detail::adaptable_bounds type-traits class. Can be replaced with constexpr if in c++17. Uses SFINAE in c++14.

Definition at line 99 of file BoundsAdaptedDist.h.

Referenced by prior_hessian::detail::make_adapted_bounded_dist_tuple(), make_adapted_bounded_dist_tuple(), prior_hessian::CompositeDist::make_component_dist(), prior_hessian::CompositeDist::make_component_dist_tuple(), and make_copula_dist().

Here is the caller graph for this function:



6.1.2.7 template<class Dist , typename = meta::EnableIfIsNotTupleT<Dist>> std::enable_if_t< !detail::DistTraitsT<Dist>::adaptable_bounds, BoundsAdaptedDistT<Dist> > prior_hessian::make_adapted_bounded_dist (Dist && *dist*)

Definition at line 104 of file BoundsAdaptedDist.h.

6.1.2.8 template<class Dist , class Vec , typename = meta::EnableIfIsNotTupleT<Dist>> std::enable_if_t<detail::DistTraitsT<Dist>::adaptable_bounds, Dist> prior_hessian::make_adapted_bounded_dist (Dist && *dist,* Vec && *lbound,* Vec && *ubound*)

Definition at line 109 of file BoundsAdaptedDist.h.

6.1.2.9 template<class Dist , class Vec , typename = meta::EnableIfIsNotTupleT<Dist>> std::enable_if_t<!detail::DistTraitsT<Dist>::adaptable_bounds, BoundsAdaptedDistT<Dist> > prior_hessian::make_adapted_bounded_dist (Dist && *dist,* Vec && *lbound,* Vec && *ubound*)

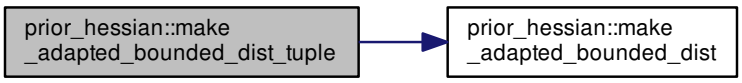
Definition at line 117 of file BoundsAdaptedDist.h.

6.1.2.10 template<class... Ts> std::tuple<BoundsAdaptedDistT<Ts>...> prior_hessian::make_adapted_bounded_dist_tuple (Ts &&... *ts*)

Definition at line 132 of file BoundsAdaptedDist.h.

References make_adapted_bounded_dist().

Here is the call graph for this function:

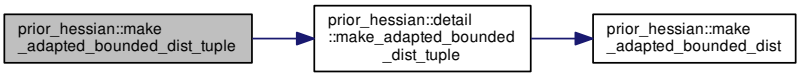


6.1.2.11 template<class... Ts> std::tuple<BoundsAdaptedDistT<Ts>...> prior_hessian::make_adapted_bounded_dist_tuple (std::tuple< Ts... > && *dists*)

Definition at line 139 of file BoundsAdaptedDist.h.

References prior_hessian::detail::make_adapted_bounded_dist_tuple().

Here is the call graph for this function:

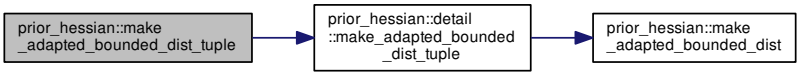


6.1.2.12 template<class... Ts> std::tuple<BoundsAdaptedDistT<Ts>...> prior_hessian::make_adapted_bounded_dist_tuple (const std::tuple< Ts... > & *dists*)

Definition at line 146 of file BoundsAdaptedDist.h.

References prior_hessian::detail::make_adapted_bounded_dist_tuple().

Here is the call graph for this function:



6.1.2.13 TruncatedGammaDist prior_hessian::make_bounded_gamma_dist (double *scale,* double *shape,* std::pair< double, double > *bounds*) [inline]

Definition at line 18 of file TruncatedGammaDist.h.

6.1.2.14 template<IdxT Ndim, class Vec , class Mat , class Vec2 > TruncatedMultivariateNormalDist<Ndim> prior_hessian::make_bounded_multivariate_normal_dist (Vec && *mu,* Mat && *sigma,* Vec2 && *lbound,* Vec2 && *ubound*)

Definition at line 22 of file TruncatedMultivariateNormalDist.h.

6.1.2.15 TruncatedNormalDist prior_hessian::make_bounded_normal_dist (double *mu,* double *sigma,* std::pair< double, double > *bounds*) [inline]

Definition at line 20 of file TruncatedNormalDist.h.

6.1.2.16 TruncatedParetoDist prior_hessian::make_bounded_pareto_dist (double *alpha,* std::pair< double, double > *bounds*) [inline]

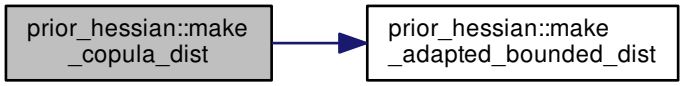
Definition at line 18 of file TruncatedParetoDist.h.

6.1.2.17 template<template< int > class CopulaTemplate, class... MarginalDistTs> CopulaDist<CopulaTemplate,MarginalDistTs...> prior_hessian::make_copula_dist (CopulaTemplate< sizeof...(MarginalDistTs)> && *copula,* MarginalDistTs &&... *dists*)

Definition at line 187 of file CopulaDist.h.

References make_adapted_bounded_dist().

Here is the call graph for this function:



6.1.2.18 ScaledSymmetricBetaDist prior_hessian::make_scaled_symmetric_beta_dist (double *beta,* std::pair< double, double > *bounds*) [inline]

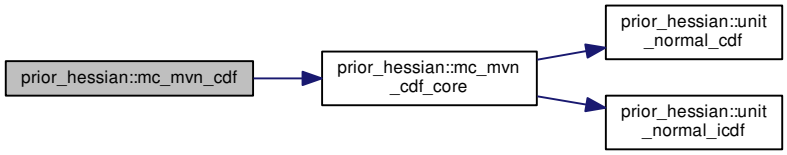
Definition at line 18 of file ScaledSymmetricBetaDist.h.

6.1.2.19 template<class Vec , class Mat > double prior_hessian::mc_mvn_cdf (const Vec & *b,* const Mat & *S,* double & *error*)

Definition at line 169 of file mvn_cdf.h.

References mc_mvn_cdf_core().

Here is the call graph for this function:



6.1.2.20 template<class Vec , class Mat > double prior_hessian::mc_mvn_cdf_core (const Vec & *b,* const Mat & *U,* double & *error,* int & *niter*)

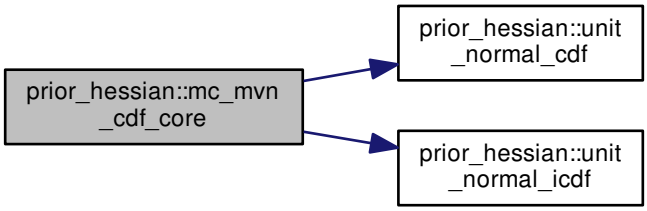
For the cdf a=-Infinity, so d=0.

Definition at line 130 of file mvn_cdf.h.

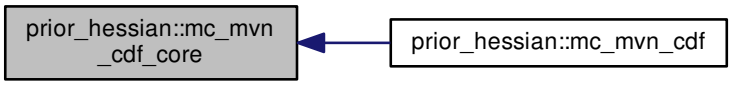
References unit_normal_cdf(), and unit_normal_icdf().

Referenced by mc_mvn_cdf().

Here is the call graph for this function:



Here is the caller graph for this function:



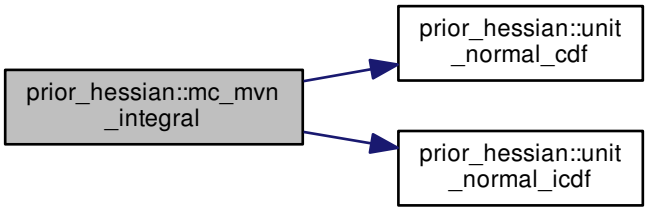
6.1.2.21 template<class Vec , class Mat > double prior_hessian::mc_mvn_integral (const Vec & *a,* const Vec & *b,* const Mat & *U,* double & *error,* int & *niter*)

compute the multivariate normal cdf integral

Definition at line 81 of file mvn_cdf.h.

References unit_normal_cdf(), and unit_normal_icdf().

Here is the call graph for this function:



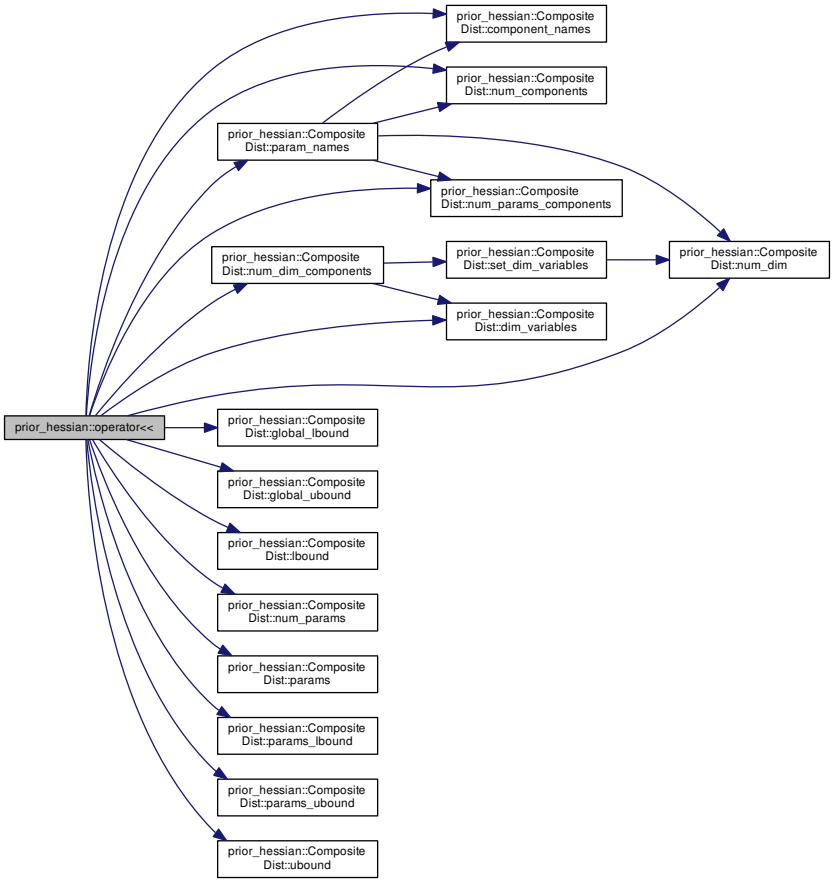
6.1.2.22 std::ostream & prior_hessian::operator<< (std::ostream & *out,* const CompositeDist & *comp_dist*)

Definition at line 80 of file CompositeDist.cpp.

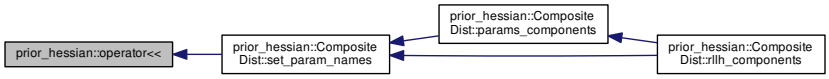
References prior_hessian::CompositeDist::component_names(), prior_hessian::CompositeDist::dim_variables(), prior_hessian::CompositeDist::global_lbound(), prior_hessian::CompositeDist::global_ubound(), prior_hessian::↩ CompositeDist::lbound(), prior_hessian::CompositeDist::num_components(), prior_hessian::CompositeDist::num_↩ dim(), prior_hessian::CompositeDist::num_dim_components(), prior_hessian::CompositeDist::num_params(), prior_↩ hessian::CompositeDist::num_params_components(), prior_hessian::CompositeDist::param_names(), prior_hessian↩ ::CompositeDist::params(), prior_hessian::CompositeDist::params_lbound(), prior_hessian::CompositeDist::params_↩ ubound(), and prior_hessian::CompositeDist::ubound().

Referenced by prior_hessian::CompositeDist::set_param_names().

Here is the call graph for this function:



Here is the caller graph for this function:



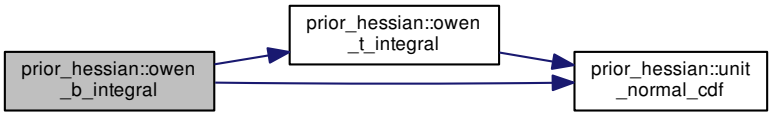
6.1.2.23 double prior_hessian::owen_b_integral (double *h*, double *k*, double *r*)

Definition at line 108 of file mvn_cdf.cpp.

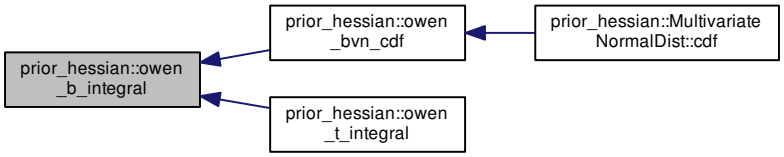
References owen_t_integral(), and unit_normal_cdf().

Referenced by owen_bvn_cdf(), and owen_t_integral().

Here is the call graph for this function:



Here is the caller graph for this function:



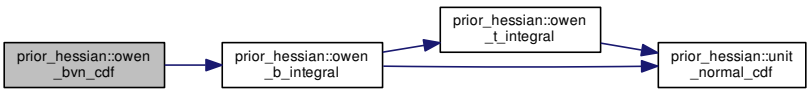
6.1.2.24 template<class Vec , class Mat > double prior_hessian::owen_bvn_cdf (const Vec & *b,* const Mat & *sigma*)

Definition at line 67 of file mvn_cdf.h.

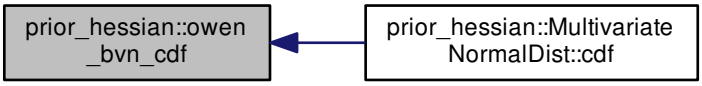
References owen_b_integral().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::cdf().

Here is the call graph for this function:



Here is the caller graph for this function:

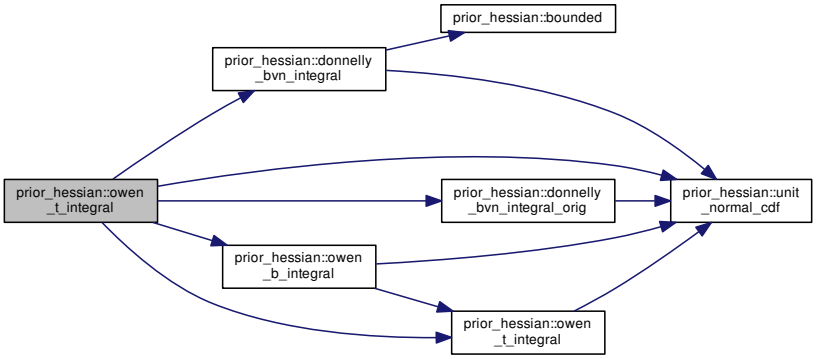


6.1.2.25 double prior_hessian::owen_t_integral (double *h,* double *a*) [inline]

Definition at line 26 of file mvn_cdf.h.

References donnelly_bvn_integral(), donnelly_bvn_integral_orig(), owen_b_integral(), owen_t_integral(), and unit_↩ normal_cdf().

Here is the call graph for this function:



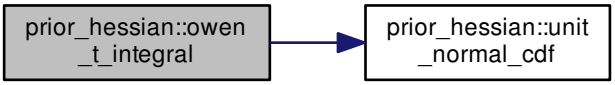
6.1.2.26 double prior_hessian::owen_t_integral (double *h,* double *a,* double *gh*)

Definition at line 54 of file mvn_cdf.cpp.

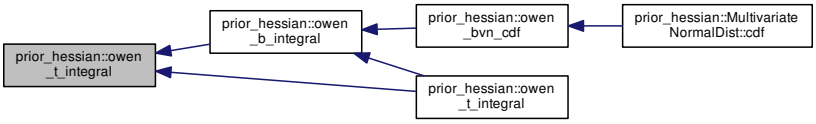
References unit_normal_cdf().

Referenced by owen_b_integral(), and owen_t_integral().

Here is the call graph for this function:



Here is the caller graph for this function:

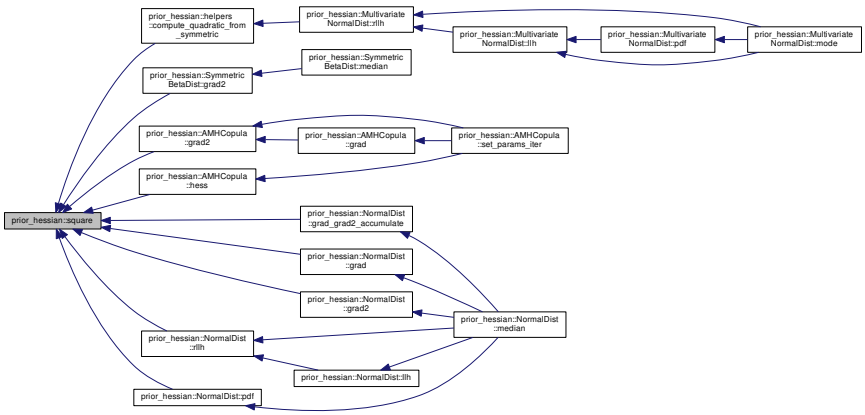


6.1.2.27 template<class T > T prior_hessian::square (T *t*)

Definition at line 44 of file util.h.

Referenced by prior_hessian::helpers::compute_quadratic_from_symmetric(), prior_hessian::NormalDist::grad(), prior_hessian::SymmetricBetaDist::grad2(), prior_hessian::NormalDist::grad2(), prior_hessian::AMHCopula< Ndim >::grad2(), prior_hessian::NormalDist::grad_grad2_accumulate(), prior_hessian::AMHCopula< Ndim >::hess(), prior_hessian::NormalDist::pdf(), and prior_hessian::NormalDist::rllh().

Here is the caller graph for this function:



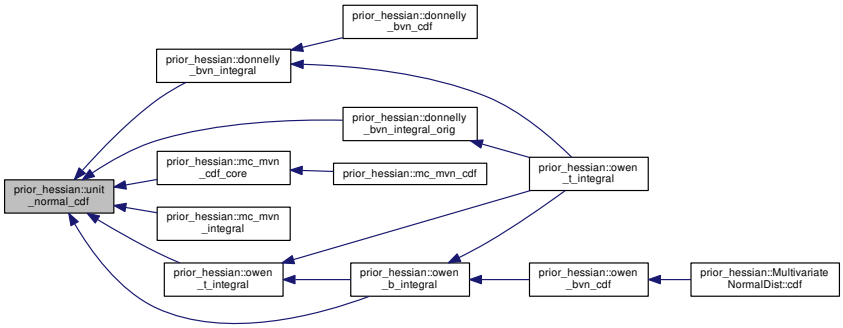
6.1.2.28 double prior_hessian::unit_normal_cdf (double *t*)

area of the lower tail of the unit normal curve below t.

Definition at line 28 of file mvn_cdf.cpp.

Referenced by donnelly_bvn_integral(), donnelly_bvn_integral_orig(), mc_mvn_cdf_core(), mc_mvn_integral(), owen_b_integral(), and owen_t_integral().

Here is the caller graph for this function:



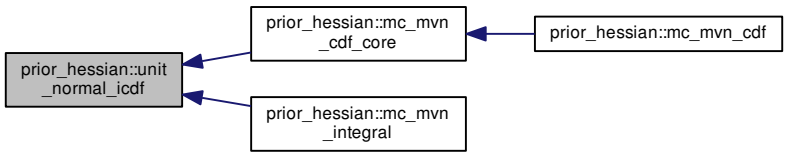
6.1.2.29 double prior_hessian::unit_normal_icdf (double *u*)

Definition at line 35 of file mvn_cdf.cpp.

References prior_hessian::constants::sqrt2.

Referenced by mc_mvn_cdf_core(), and mc_mvn_integral().

Here is the caller graph for this function:



6.2 prior_hessian::constants Namespace Reference

Variables

- const double sqrt2 = std::sqrt(2.)
- const double sqrt2_inv = 1./std::sqrt(2.)
- const double sqrt2pi = std::sqrt(2.∗arma::datum::pi)
- const double sqrt2pi_inv = 1./std::sqrt(2.∗arma::datum::pi)
- const double log2pi = std::log(2.∗arma::datum::pi)

6.2.1 Variable Documentation

6.2.1.1 const double prior_hessian::constants::log2pi = std::log(2.∗arma::datum::pi)

Definition at line 16 of file util.cpp.

Referenced by prior_hessian::NormalDist::llh(), and prior_hessian::MultivariateNormalDist< Ndim >::sample().

6.2.1.2 const double prior_hessian::constants::sqrt2 = std::sqrt(2.)

Definition at line 12 of file util.cpp.

Referenced by prior_hessian::NormalDist::icdf(), and prior_hessian::unit_normal_icdf().

6.2.1.3 const double prior_hessian::constants::sqrt2_inv = 1./std::sqrt(2.)

Definition at line 13 of file util.cpp.

Referenced by prior_hessian::NormalDist::cdf().

6.2.1.4 const double prior_hessian::constants::sqrt2pi = std::sqrt(2.∗arma::datum::pi)

Definition at line 14 of file util.cpp.

6.2.1.5 const double prior_hessian::constants::sqrt2pi_inv = 1./std::sqrt(2.∗arma::datum::pi)

Definition at line 15 of file util.cpp.

Referenced by prior_hessian::NormalDist::pdf().

6.3 prior_hessian::CopulaDistImpl Namespace Reference

Classes

- class CopulaDist

6.4 prior_hessian::detail Namespace Reference

Classes

- class dist_adaptor_traits
- struct dist_adaptor_traits< CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... > >
- struct dist_adaptor_traits< GammaDist >
- struct dist_adaptor_traits< MultivariateNormalDist< Ndim > >
- struct dist_adaptor_traits< NormalDist >
- struct dist_adaptor_traits< ParetoDist >
- struct dist_adaptor_traits< ScaledSymmetricBetaDist >
- struct dist_adaptor_traits< SymmetricBetaDist >
- struct dist_adaptor_traits< TruncatedGammaDist >
- struct dist_adaptor_traits< TruncatedMultivariateNormalDist< Ndim > >
- struct dist_adaptor_traits< TruncatedNormalDist >
- struct dist_adaptor_traits< TruncatedParetoDist >

Typedefs

- template<class DistT >
  using DistTraitsT = detail::dist_adaptor_traits< std::decay_t< DistT >>

Functions

- template<class... Ts, std::size_t... I>
  std::tuple< BoundsAdaptedDistT< Ts >... > make_adapted_bounded_dist_tuple (std::tuple< Ts... > &&dists, std::index_sequence< I... >)
- template<class... Ts, std::size_t... I>
  std::tuple< BoundsAdaptedDistT< Ts >... > make_adapted_bounded_dist_tuple (const std::tuple< Ts... > &dists, std::index_sequence< I... >)
- template<long N, long... I>
  VecT eulerian_polynomial ()

6.4.1 Typedef Documentation

6.4.1.1 template<class DistT > using prior_hessian::detail::DistTraitsT = typedef detail::dist_adaptor_traits<std::decay_t<DistT>>

Type traits class for distribution type DistT.

The traits class describes the Adaptor classes applicable to each individual distribution

Definition at line 52 of file BoundsAdaptedDist.h.

6.4.2 Function Documentation

6.4.2.1 template<long N, long... I> VecT prior_hessian::detail::eulerian_polynomial ( )

Definition at line 26 of file EulerianPolynomial.h.

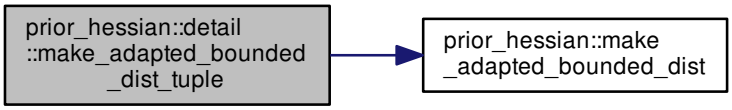
6.4.2.2 template<class... Ts, std::size_t... I> std::tuple<BoundsAdaptedDistT<Ts>...> prior_hessian::detail::make_adapted_bounded_dist_tuple (std::tuple< Ts... > && *dists*, std::index_sequence< I... >)

Definition at line 66 of file BoundsAdaptedDist.h.

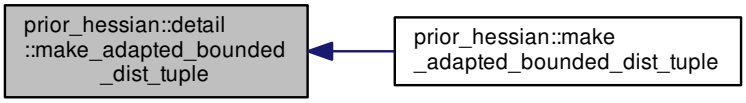
References prior_hessian::make_adapted_bounded_dist().

Referenced by prior_hessian::make_adapted_bounded_dist_tuple().

Here is the call graph for this function:



Here is the caller graph for this function:

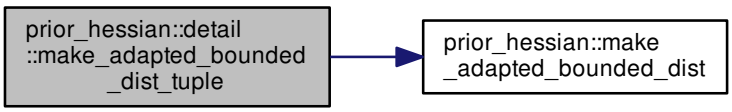


6.4.2.3 template<class... Ts, std::size_t... I> std::tuple<BoundsAdaptedDistT<Ts>...> prior_hessian::detail::make_adapted_bounded_dist_tuple (const std::tuple< Ts... > & *dists*, std::index_sequence< I... >)

Definition at line 73 of file BoundsAdaptedDist.h.

References prior_hessian::make_adapted_bounded_dist().

Here is the call graph for this function:



6.5 prior_hessian::genz Namespace Reference

Namespaces

- fortran

Functions

- template<class Vec , class Mat >
  double mvn_cdf_genz (const Vec &b, const Mat &S, double &error)

6.5.1 Function Documentation

6.5.1.1 template<class Vec , class Mat > double prior_hessian::genz::mvn_cdf_genz (const Vec & *b*, const Mat & *S*, double & *error*)

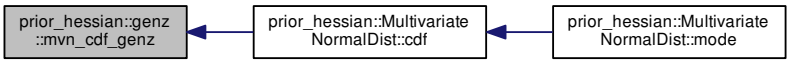
Definition at line 217 of file mvn_cdf.h.

References prior_hessian::genz::fortran::mvndst_().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::cdf().

Here is the call graph for this function:

Here is the caller graph for this function:



6.6 prior_hessian::genz::fortran Namespace Reference

Functions

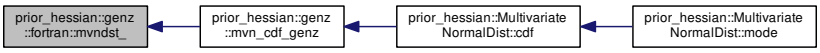
- int mvndst_ (int *n, double lower[ ], double upper[ ], int infin[ ], double correl[ ], int *maxpts, double *abseps, double *releps, double *error, double *value, int *inform)

6.6.1 Function Documentation

6.6.1.1 int prior_hessian::genz::fortran::mvndst_ (int * *n*, double *lower[ ]*, double *upper[ ]*, int *infin[ ]*, double *correl[ ]*, int * *maxpts*, double * *abseps*, double * *releps*, double * *error*, double * *value*, int * *inform*)

Referenced by prior_hessian::genz::mvn_cdf_genz().

Here is the caller graph for this function:



6.7 prior_hessian::helpers Namespace Reference

Functions

- template<class Vec , class Mat >
  double compute_quadratic_from_symmetric (IdxT Ndim, const Vec &v, const Mat &A)

6.7.1 Function Documentation

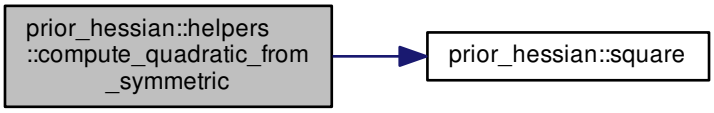
6.7.1.1 template<class Vec , class Mat > double prior_hessian::helpers::compute_quadratic_from_symmetric (IdxT *Ndim*, const Vec & *v*, const Mat & *A*)

Definition at line 137 of file MultivariateNormalDist.h.

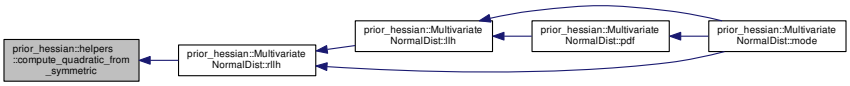
References prior_hessian::square().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::rllh().

Here is the call graph for this function:



Here is the caller graph for this function:



6.8 prior_hessian::mcmc Namespace Reference

Classes

- class MCMCData

6.9 prior_hessian::meta Namespace Reference

Class templates to utilize sequencing behavior of std::initializer_list expressions.

Classes

- struct all_dists_are_bounded
- struct conjunction
- struct conjunction< B1 >
- struct conjunction< B1, Bn... >
- struct disjunction
- struct disjunction< B1 >
- struct disjunction< B1, Bn... >
- class is_copula
- struct is_numeric_template_of
- struct is_numeric_template_of< ClassNumericTemplate, ClassNumericTemplate< Is... > >
- class is_subclass_of_numeric_template
- struct is_template_of
- struct is_template_of< ClassTemplate, ClassTemplate< Ts... > >

Typedefs

- template<class... DistTs>
  using ConstructableIfAllDistsAreBoundedT = std::enable_if< all_dists_are_bounded< DistTs... >::value, bool >
- template<class... DistTs>
  using ConstructableIfNotAllDistsAreBoundedT = std::enable_if<!all_dists_are_bounded< DistTs... >::value, bool >
- template<template< template< int > class, class... > class CopulaT, class U >
  using ConstructableIfIsCopulaT = std::enable_if_t< is_copula< CopulaT, U >::value,bool >
- template<class ReturnT , class BoolT >
  using ReturnIfT = std::enable_if_t< BoolT::value, ReturnT >
- template<bool val>
  using ConstructableIf = std::enable_if_t< val, bool >
- template<bool val>
  using ConstructableIfNot = std::enable_if_t<!val, bool >
- template<class T , class SelfT >
  using ConstructableIfNotSelfT = std::enable_if_t<!std::is_same< std::decay_t< T >, SelfT >::value, bool >
- template<class T , class BaseT >
  using EnableIfSubclassT = std::enable_if_t< std::is_base_of< std::remove_reference_t< BaseT >, std::remove_reference_t< T >>::value >
- template<class T , template< int > class ClassNumericTemplate>
  using EnableIfSubclassOfNumericTemplateT = std::enable_if_t< is_subclass_of_numeric_template< ClassNumericTemplate, std::remove_reference_t< T >>::value >
- template<class ReturnT , class T , template< int > class ClassNumericTemplate>
  using ReturnIfSubclassOfNumericTemplateT = std::enable_if_t< is_subclass_of_numeric_template< ClassNumericTemplate, std::remove_reference_t< T >>::value, ReturnT >
- template<class T , class SelfT >
  using EnableIfNotIsSelfT = std::enable_if_t< !std::is_same< std::decay_t< T >, SelfT >::value >
- template<class ReturnT , class T , class BaseT >
  using ReturnIfSubclassT = std::enable_if_t< std::is_base_of< std::remove_reference_t< BaseT >, std::remove_reference_t< T >>::value, ReturnT >
- template<class BaseT , class... Ts>
  using EnableIfIsSuperclassOfAllT = std::enable_if_t< conjunction< std::is_base_of< std::remove_reference_t< BaseT >, std::remove_reference_t< Ts >>... >::value >
- template<class T , template< typename... > class ClassTemplate>
  using EnableIfInstantiatedFromT = std::enable_if_t< is_template_of< ClassTemplate, std::remove_reference_t< T >>::value >
- template<class T , template< int > class ClassTemplate>
  using EnableIfInstantiatedFromNumericT = std::enable_if_t< is_numeric_template_of< ClassTemplate, std::remove_reference_t< T >>::value >
- template<class ReturnT , class T , template< int > class ClassTemplate>
  using ReturnIfInstantiatedFromNumericT = std::enable_if_t< is_numeric_template_of< ClassTemplate, std::remove_reference_t< T >>::value, ReturnT >
- template<class T , template< typename... > class ClassTemplate>
  using EnableIfNotInstantiatedFromT = std::enable_if_t< !is_template_of< ClassTemplate, std::remove_reference_t< T >>::value >
- template<class ReturnT , class TestT , template< typename... > class ClassTemplate>
  using ReturnIfInstantiatedFromT = std::enable_if_t< is_template_of< ClassTemplate, std::remove_reference_t< TestT >>::value, ReturnT >
- template<class ReturnT , class TestT , template< typename... > class ClassTemplate>
  using ReturnIfNotInstantiatedFromT = std::enable_if_t< !is_template_of< ClassTemplate, std::remove_reference_t< TestT >>::value, ReturnT >

- template<template< typename > class ClassTemplate, class... Ts>
  using EnableIfIsTemplateForAllT = std::enable_if_t< conjunction< is_template_of< ClassTemplate, std::remove_reference_t< Ts >>... >::value >
- template<template< typename... > class ClassTemplate, class... Ts>
  using ConstructableIfIsTemplateForAllT = std::enable_if_t< conjunction< is_template_of< ClassTemplate, std::remove_reference_t< Ts >>... >::value, bool >
- template<class SuperClass , class T >
  using ConstructableIfIsSuperClassT = std::enable_if_t< std::is_base_of< std::remove_reference_t< SuperClass >, std::remove_reference_t< T >>::value, bool >
- template<class SuperClass , class... Ts>
  using ConstructableIfIsSuperClassForAllT = std::enable_if_t< conjunction< std::is_base_of< std::remove_reference_t< SuperClass >, std::remove_reference_t< Ts >>... >::value, bool >
- template<class T , template< int > class ClassTemplate>
  using ConstructableIfInstantiatedFromNumericT = std::enable_if_t< is_numeric_template_of< ClassTemplate, std::remove_reference_t< T >>::value, bool >
- template<class T >
  using EnableIfIsNotTupleT = std::enable_if_t< !is_template_of< std::tuple, std::remove_reference_t< T >>::value >
- template<class... Ts>
  using EnableIfNonEmpty = std::enable_if_t< (sizeof...(Ts)>0) >
- template<class... Ts>
  using EnableIfAllAreNotTupleT = std::enable_if_t< !disjunction< is_template_of< std::tuple, std::remove_reference_t< Ts >>... >::value >
- template<class SelfT , class T >
  using EnableIfIsNotTupleAndIsNotSelfT = std::enable_if_t< !is_template_of< std::tuple, std::remove_reference_t< T >>::value &&!std::is_same< std::decay_t< T >, SelfT >::value >
- template<class T , class... Ts>
  using ConstructableIfAllAreNotTupleAndAreNotT = std::enable_if_t< !disjunction< is_template_of< std::tuple, std::remove_reference_t< Ts >>... >::value &&!disjunction< std::is_same< std::decay_t< Ts >, T >... >::value, bool >
- template<class Dist , class BaseDist >
  using DerivedFrom = std::enable_if_t< std::is_base_of< std::decay_t< BaseDist >, std::decay_t< Dist >>::value, std::decay_t< Dist >>

Functions

- template<class T = int>
  void call_in_order (std::initializer_list< T >)
- template<class InputIterator , class ResultT , class BinaryOperation >
  PRIOR_HESSIAN_META_CONSTEXPR ResultT constexpr_accumulate (InputIterator first, InputIterator last, ResultT init, BinaryOperation op)
- PRIOR_HESSIAN_META_CONSTEXPR bool logical_and_in_order (std::initializer_list< bool > L)
- template<class T >
  PRIOR_HESSIAN_META_CONSTEXPR T sum_in_order (std::initializer_list< T > L)
- template<class T >
  PRIOR_HESSIAN_META_CONSTEXPR T prod_in_order (std::initializer_list< T > L)

6.9.1 Detailed Description

Class templates to utilize sequencing behavior of std::initializer_list expressions.

These class templates are intended to be used in variadic template functions to sequence the order of calls as a std::initializer_list.

6.9.2 Typedef Documentation

6.9.2.1 template<bool val> using prior_hessian::meta::ConstructableIf = typedef std::enable_if_t<val,bool>

Definition at line 117 of file Meta.h.

6.9.2.2 template<class T , class... Ts> using prior_hessian::meta::ConstructableIfAllAreNotTupleAndAreNotT = typedef std::enable_if_t< !disjunction<is_template_of<std::tuple,std::remove_reference_t<Ts>>...>::value && !disjunction<std::is_same<std::decay_t<Ts>,T>...>::value, bool>

Definition at line 214 of file Meta.h.

6.9.2.3 template<class... DistTs> using prior_hessian::meta::ConstructableIfAllDistsAreBoundedT = typedef std::enable_if<all_dists_are_bounded<DistTs...>::value,bool>

Definition at line 87 of file BoundsAdaptedDist.h.

6.9.2.4 template<class T , template< int > class ClassTemplate> using prior_hessian::meta::ConstructableIfInstantiatedFromNumericT = typedef std::enable_if_t< is_numeric_template_of<ClassTemplate, std::remove_reference_t<T>>::value, bool >

Definition at line 194 of file Meta.h.

6.9.2.5 template<template< template< int > class, class... > class CopulaT, class U > using prior_hessian::meta::ConstructableIfIsCopulaT = typedef std::enable_if_t<is_copula<CopulaT,U>::value ,bool>

Definition at line 111 of file Meta.h.

6.9.2.6 template<class SuperClass , class... Ts> using prior_hessian::meta::ConstructableIfIsSuperClassForAllT = typedef std::enable_if_t< conjunction< std::is_base_of<std::remove_reference_t<SuperClass>,std::remove_reference_t<Ts>>... >::value, bool>

Definition at line 190 of file Meta.h.

6.9.2.7 template<class SuperClass , class T > using prior_hessian::meta::ConstructableIfIsSuperClassT = typedef std::enable_if_t< std::is_base_of<std::remove_reference_t<SuperClass>,std::remove_reference_t<T>>::value, bool>

Definition at line 186 of file Meta.h.

6.9.2.8 template<template< typename... > class ClassTemplate, class... Ts> using prior_hessian::meta::ConstructableIfIsTemplateForAllT = typedef std::enable_if_t< conjunction< is_template_of<ClassTemplate,std::remove_reference_t<Ts>> ... >::value, bool>

Definition at line 182 of file Meta.h.

6.9.2.9 template<bool val> using prior_hessian::meta::ConstructableIfNot = typedef std::enable_if_t<!val,bool>

Definition at line 120 of file Meta.h.

6.9.2.10 template<class... DistTs> using prior_hessian::meta::ConstructableIfNotAllDistsAreBoundedT = typedef std::enable_if<!all_dists_are_bounded<DistTs...>::value,bool>

Definition at line 89 of file BoundsAdaptedDist.h.

6.9.2.11 template<class T , class SelfT > using prior_hessian::meta::ConstructableIfNotSelfT = typedef std::enable_if_t<!std::is_same<std::decay_t<T>,SelfT>::value,bool>

Definition at line 123 of file Meta.h.

6.9.2.12 template<class Dist , class BaseDist > using prior_hessian::meta::DerivedFrom = typedef std::enable_if_t<std::is_base_of<std::decay_t<BaseDist>,std::decay_t<Dist>>::value,std::decay_t<Dist>>

Definition at line 228 of file Meta.h.

6.9.2.13 template<class... Ts> using prior_hessian::meta::EnableIfAllAreNotTupleT = typedef std::enable_if_t< !disjunction< is_template_of<std::tuple,std::remove_reference_t<Ts>>... >::value >

Definition at line 204 of file Meta.h.

6.9.2.14 template<class T , template< int > class ClassTemplate> using prior_hessian::meta::EnableIf↩ InstantiatedFromNumericT = typedef std::enable_if_t< is_numeric_template_of<ClassTemplate, std::remove_reference_t<T>>::value>

Definition at line 158 of file Meta.h.

6.9.2.15 template<class T , template< typename... > class ClassTemplate> using prior_hessian::meta↩ ::EnableIfInstantiatedFromT = typedef std::enable_if_t< is_template_of<ClassTemplate, std::remove_reference_t<T>>::value >

Definition at line 150 of file Meta.h.

6.9.2.16 template<class SelfT , class T > using prior_hessian::meta::EnableIfIsNotTupleAndIsNotSelfT = typedef std::enable_if_t< !is_template_of<std::tuple,std::remove_reference_t<T>>::value && !std::is_same<std::decay_t<T>,SelfT>::value >

Definition at line 209 of file Meta.h.

6.9.2.17 template<class T > using prior_hessian::meta::EnableIfIsNotTupleT = typedef std::enable_if_t< !is_template_of<std::tuple,std::remove_reference_t<T>>::value >

Definition at line 197 of file Meta.h.

6.9.2.18 template<class BaseT , class... Ts> using prior_hessian::meta::EnableIfIsSuperclassOfAllT = typedef std::enable_if_t<conjunction< std::is_base_of<std::remove_reference_t<BaseT>,std::remove_reference_t<Ts>> ... >::value >

Definition at line 146 of file Meta.h.

6.9.2.19 template<template< typename > class ClassTemplate, class... Ts> using prior_hessian::meta::EnableIfIsTemplateForAllT = typedef std::enable_if_t< conjunction< is_template_of<ClassTemplate,std::remove_reference_t<Ts>> ... >::value >

Definition at line 178 of file Meta.h.

6.9.2.20 template<class... Ts> using prior_hessian::meta::EnableIfNonEmpty = typedef std::enable_if_t< (sizeof...(Ts)>0) >

Definition at line 200 of file Meta.h.

6.9.2.21 template<class T , template< typename... > class ClassTemplate> using prior_hessian::meta::EnableIfNotInstantiatedFromT = typedef std::enable_if_t< !is_template_of<ClassTemplate, std::remove_reference_t<T>>::value >

Definition at line 166 of file Meta.h.

6.9.2.22 template<class T , class SelfT > using prior_hessian::meta::EnableIfNotIsSelfT = typedef std::enable_if_t< !std::is_same<std::decay_t<T>,SelfT>::value >

Definition at line 138 of file Meta.h.

6.9.2.23 template<class T , template< int > class ClassNumericTemplate> using prior_hessian::meta::EnableIfSubclassOfNumericTemplateT = typedef std::enable_if_t< is_subclass_of_numeric_template<ClassNumericTemplate, std::remove_reference_t<T>>::value>

Definition at line 131 of file Meta.h.

6.9.2.24 template<class T , class BaseT > using prior_hessian::meta::EnableIfSubclassT = typedef std::enable_if_t< std::is_base_of<std::remove_reference_t<BaseT>,std::remove_reference_t<T>>::value>

Definition at line 127 of file Meta.h.

6.9.2.25 template<class ReturnT , class T , template< int > class ClassTemplate> using prior_hessian::meta::ReturnIfInstantiatedFromNumericT = typedef std::enable_if_t< is_numeric_template_of<ClassTemplate, std::remove_reference_t<T>>::value, ReturnT>

Definition at line 162 of file Meta.h.

6.9.2.26 template<class ReturnT , class TestT , template< typename... > class ClassTemplate> using prior_hessian::meta::ReturnIfInstantiatedFromT = typedef std::enable_if_t< is_template_of<ClassTemplate, std::remove_reference_t<TestT>>::value, ReturnT>

Definition at line 170 of file Meta.h.

6.9.2.27 template<class ReturnT , class TestT , template< typename... > class ClassTemplate> using prior_hessian::meta::ReturnIfNotInstantiatedFromT = typedef std::enable_if_t< !is_template_of<ClassTemplate, std::remove_reference_t<TestT>>::value, ReturnT>

Definition at line 174 of file Meta.h.

6.9.2.28 template<class ReturnT , class T , template< int > class ClassNumericTemplate> using prior_hessian::meta::ReturnIfSubclassOfNumericTemplateT = typedef std::enable_if_t< is_subclass_of_numeric_template<ClassNumericTemplate, std::remove_reference_t<T>>::value, ReturnT>

Definition at line 135 of file Meta.h.

6.9.2.29 template<class ReturnT , class T , class BaseT > using prior_hessian::meta::ReturnIfSubclassT = typedef std::enable_if_t< std::is_base_of<std::remove_reference_t<BaseT>,std::remove_reference_t<T>>::value, ReturnT>

Definition at line 142 of file Meta.h.

6.9.2.30 template<class ReturnT , class BoolT > using prior_hessian::meta::ReturnIfT = typedef std::enable_if_t<BoolT::value,ReturnT>

Definition at line 114 of file Meta.h.

6.9.3 Function Documentation

6.9.3.1 template<class T = int> void prior_hessian::meta::call_in_order (std::initializer_list< T >)

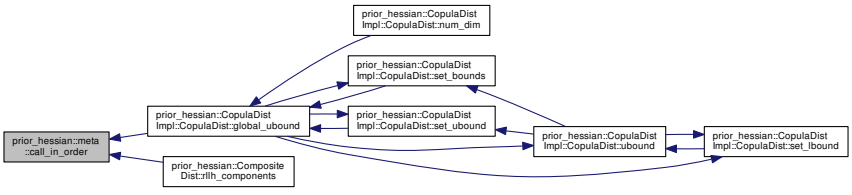
NOOP function which is used to ensure call order on a variadic sequence of function calls

Definition at line 41 of file Meta.h.

References PRIOR_HESSIAN_META_CONSTEXPR.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CompositeDist::rllh_components().

Here is the caller graph for this function:



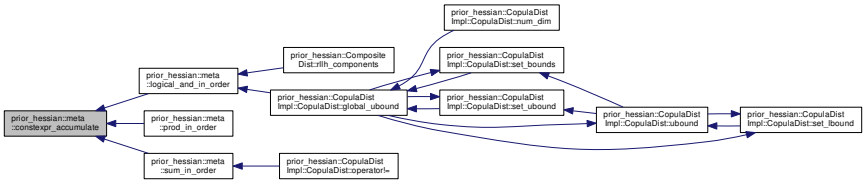
6.9.3.2 template<class InputIterator , class ResultT , class BinaryOperation > PRIOR_HESSIAN_META_CONSTEXPR ResultT prior_hessian::meta::constexpr_accumulate (InputIterator *first,* InputIterator *last,* ResultT *init,* BinaryOperation *op*)

Definition at line 46 of file Meta.h.

References PRIOR_HESSIAN_META_CONSTEXPR.

Referenced by logical_and_in_order(), prod_in_order(), and sum_in_order().

Here is the caller graph for this function:



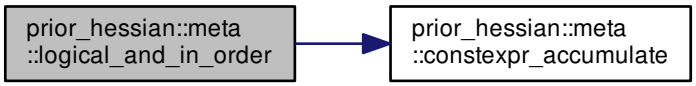
6.9.3.3 PRIOR_HESSIAN_META_CONSTEXPR bool prior_hessian::meta::logical_and_in_order (std::initializer_list< bool > *L*) [inline]

Definition at line 53 of file Meta.h.

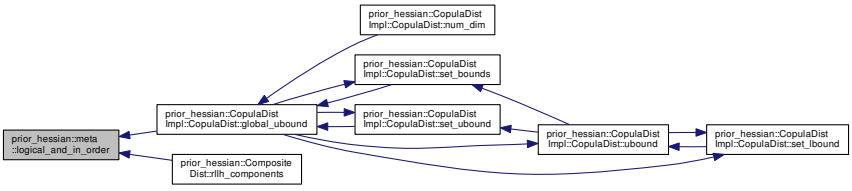
References constexpr_accumulate(), and PRIOR_HESSIAN_META_CONSTEXPR.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CompositeDist::rIlh_components().

Here is the call graph for this function:



Here is the caller graph for this function:

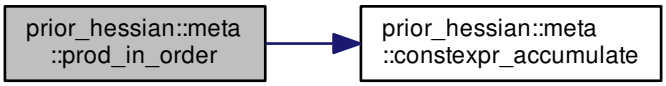


6.9.3.4 template<class T > PRIOR_HESSIAN_META_CONSTEXPR T prior_hessian::meta::prod_in_order (std::initializer_list< T > *L*)

Definition at line 63 of file Meta.h.

References constexpr_accumulate().

Here is the call graph for this function:



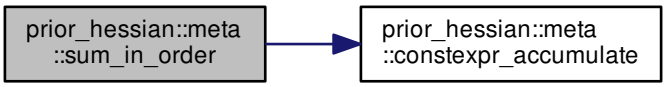
6.9.3.5 template<class T > PRIOR_HESSIAN_META_CONSTEXPR T prior_hessian::meta::sum_in_order (std::initializer_list< T > *L*)

Definition at line 58 of file Meta.h.

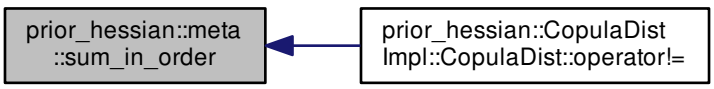
References constexpr_accumulate(), and PRIOR_HESSIAN_META_CONSTEXPR.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:



6.10 prior_hessian::polylog Namespace Reference

Functions

- template<int n>
  double eulerian_polynomial (double z)
- template<>
  double eulerian_polynomial< 0 > (double)
- template<>
  double eulerian_polynomial< 1 > (double z)
- template<>
  double eulerian_polynomial< 2 > (double z)
- template<>
  double eulerian_polynomial< 3 > (double z)
- template<>
  double eulerian_polynomial< 4 > (double z)
- template<>
  double eulerian_polynomial< 5 > (double z)
- template<>
  double eulerian_polynomial< 6 > (double z)
- template<>
  double eulerian_polynomial< 7 > (double z)
- template<>
  double eulerian_polynomial< 8 > (double z)
- template<>
  double eulerian_polynomial< 9 > (double z)
- template<int n>
  double polylog (double z)
- template<>
  double polylog< 1 > (double z)

6.10.1 Function Documentation

6.10.1.1 template<int n> double prior_hessian::polylog::eulerian_polynomial (double *z*)

6.10.1.2 template<> double prior_hessian::polylog::eulerian_polynomial< 0 > (double)

Definition at line 17 of file PolyLog.h.

6.10.1.3 template<> double prior_hessian::polylog::eulerian_polynomial< 1 > (double *z*)

Definition at line 23 of file PolyLog.h.

6.10.1.4 template<> double prior_hessian::polylog::eulerian_polynomial< 2 > (double *z*)

Definition at line 29 of file PolyLog.h.

6.10.1.5 template<> double prior_hessian::polylog::eulerian_polynomial< 3 > (double z)

Definition at line 35 of file PolyLog.h.

6.10.1.6 template<> double prior_hessian::polylog::eulerian_polynomial< 4 > (double z)

Definition at line 41 of file PolyLog.h.

6.10.1.7 template<> double prior_hessian::polylog::eulerian_polynomial< 5 > (double z)

Definition at line 47 of file PolyLog.h.

6.10.1.8 template<> double prior_hessian::polylog::eulerian_polynomial< 6 > (double z)

Definition at line 53 of file PolyLog.h.

6.10.1.9 template<> double prior_hessian::polylog::eulerian_polynomial< 7 > (double z)

Definition at line 59 of file PolyLog.h.

6.10.1.10 template<> double prior_hessian::polylog::eulerian_polynomial< 8 > (double z)

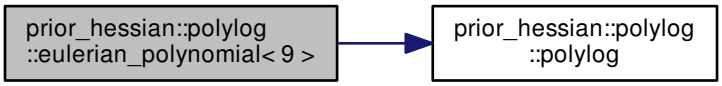
Definition at line 65 of file PolyLog.h.

6.10.1.11 template<> double prior_hessian::polylog::eulerian_polynomial< 9 > (double z)

Definition at line 70 of file PolyLog.h.

References polylog().

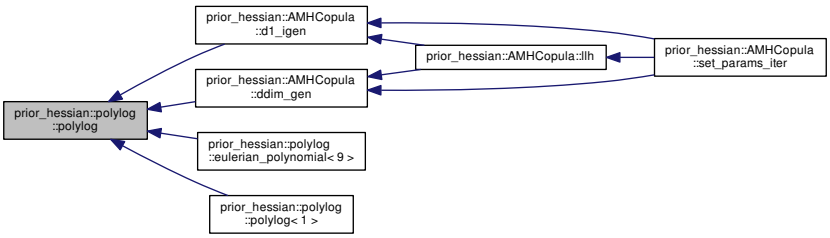
Here is the call graph for this function:



6.10.1.12 template<int n> double prior_hessian::polylog::polylog (double *z*)

Referenced by prior_hessian::AMHCopula< Ndim >::d1_igen(), prior_hessian::AMHCopula< Ndim >::ddim_gen(), eulerian_polynomial< 9 >(), and polylog< 1 >().

Here is the caller graph for this function:

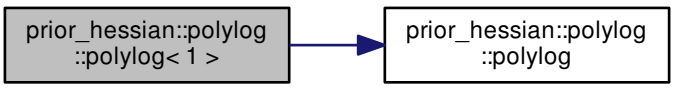


6.10.1.13 template<> double prior_hessian::polylog::polylog< 1 > (double *z*)

Definition at line 120 of file PolyLog.h.

References polylog().

Here is the call graph for this function:



7 Class Documentation

7.1 prior_hessian::meta::all_dists_are_bounded< DistTs > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ BoundsAdaptedDist.h>`

Static Public Attributes

- static constexpr bool value = meta::logical_and_in_order({detail::dist_adaptor_traits<std::decay_t<DistTs>>↩ ::value...})

7.1.1 Detailed Description

template<class... DistTs>
struct prior_hessian::meta::all_dists_are_bounded< DistTs >

Definition at line 81 of file BoundsAdaptedDist.h.

7.1.2 Member Data Documentation

7.1.2.1 template<class... DistTs> constexpr bool prior_hessian::meta::all_dists_are_bounded< DistTs >::value = meta::logical_and_in_order({detail::dist_adaptor_traits<std::decay_t<DistTs>>::value...}) [static]

Definition at line 83 of file BoundsAdaptedDist.h.

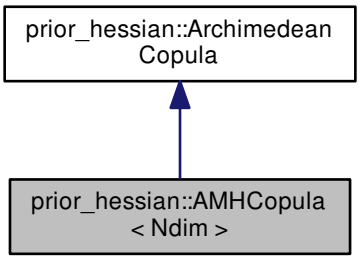
The documentation for this struct was generated from the following file:

- BoundsAdaptedDist.h

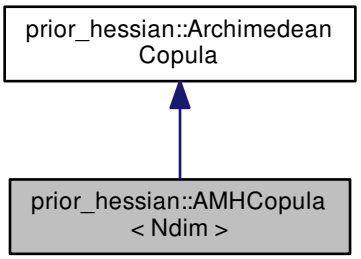
7.2 prior_hessian::AMHCopula< Ndim > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/AMH↩Copula.h>`

Inheritance diagram for prior_hessian::AMHCopula< Ndim >:



Collaboration diagram for prior_hessian::AMHCopula< Ndim >:



Public Types

- using NdimVecT = arma::Col< double >::fixed< Ndim >
- using NdimMatT = arma::Mat< double >::fixed< Ndim, Ndim >

Public Member Functions

- AMHCopula ()
- AMHCopula (double theta)
- double theta () const
- void set_theta (double val)
- bool operator== (const AMHCopula< Ndim > &o) const
- bool operator!= (const AMHCopula< Ndim > &o) const
- template<class Vec >
  void set_params (const Vec &params)
- template<class IterT >
  void append_params (IterT &params)
- template<class IterT >
  void set_params_iter (IterT &params)
- template<class Vec >
  double cdf (const Vec &u) const
- template<class Vec >
  double pdf (const Vec &u) const
- template<class Vec >
  double llh (const Vec &u) const
- template<class Vec >
  double rllh (const Vec &u) const
- double rllh_const () const
- template<class Vec >
  NdimVecT grad (const Vec &u) const
- template<class Vec >
  NdimVecT grad2 (const Vec &u) const

- template<class Vec >
  NdimMatT hess (const Vec &u) const
- template<class Vec , class Vec2 >
  void rllh_grad_accumulate (const Vec &u, double &rllh, Vec2 &grad) const
- template<class Vec , class Vec2 >
  void rllh_grad_grad2_accumulate (const Vec &u, double &rllh, Vec2 &grad, Vec2 &grad2) const
- template<class Vec , class Vec2 , class Mat >
  void rllh_grad_hess_accumulate (const Vec &u, double &rllh, Vec2 &grad, Mat &hess) const
- template<class RngT >
  NdimVecT sample (RngT &rng) const
- double gen (double t) const
- double ddim_gen (double t) const
- double igen (double u) const
- double d1_igen (double u) const
- template<class Vec >
  double igen_sum (const Vec &u)

Static Public Member Functions

- static const StringVecT & param_names ()
- static constexpr IdxT num_params ()
- static constexpr IdxT num_dim ()
- static double param_lbound ()
- static double param_ubound ()
- static bool check_theta (double val)
- template<class Vec >
  static bool check_params (const Vec &params)
- template<class IterT >
  static bool check_params_iter (IterT &params)
- template<class Vec >
  static void rllh_dtheta_accumulate (double theta, const Vec &u, double &rllh, double &dtheta)
- template<class Vec >
  static void rllh_d2theta_accumulate (double theta, const Vec &u, double &rllh, double &dtheta, double &d2theta)

7.2.1 Detailed Description

template<int Ndim>
class prior_hessian::AMHCopula< Ndim >

Definition at line 18 of file AMHCopula.h.

7.2.2 Member Typedef Documentation

7.2.2.1 template<int Ndim> using prior_hessian::AMHCopula< Ndim >::NdimMatT = arma::Mat<double>::fixed<Ndim,Ndim>

Definition at line 27 of file AMHCopula.h.

7.2.2.2 template<int Ndim> using prior_hessian::AMHCopula< Ndim >::NdimVecT = arma::Col<double>::fixed<Ndim>

Definition at line 26 of file AMHCopula.h.

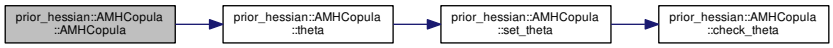
7.2.3 Constructor & Destructor Documentation

7.2.3.1 template<int Ndim> prior_hessian::AMHCopula< Ndim >::AMHCopula ( ) [inline]

Definition at line 35 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::theta().

Here is the call graph for this function:

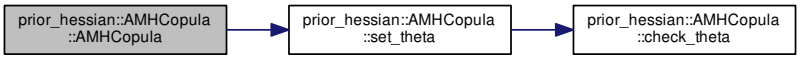


7.2.3.2 template<int Ndim> prior_hessian::AMHCopula< Ndim >::AMHCopula (double *theta*)

Definition at line 169 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::set_theta().

Here is the call graph for this function:



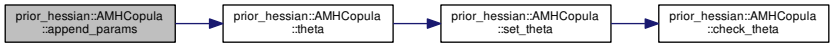
7.2.4 Member Function Documentation

7.2.4.1 template<int Ndim> template<class IterT > void prior_hessian::AMHCopula< Ndim >::append_params (IterT & *params*) [inline]

Definition at line 51 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::theta().

Here is the call graph for this function:



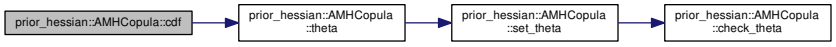
7.2.4.2 template<int Ndim> template<class Vec > double prior_hessian::AMHCopula< Ndim >::cdf (const Vec & *u*) const

Definition at line 194 of file AMHCopula.h.

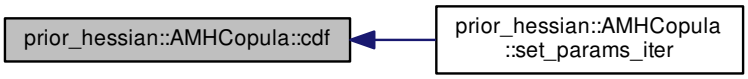
References prior_hessian::AMHCopula< Ndim >::theta().

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:

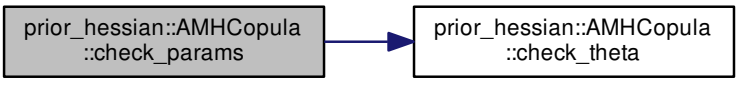


7.2.4.3 template<int Ndim> template<class Vec > static bool prior_hessian::AMHCopula< Ndim >::check_params (const Vec & *params*) [inline], [static]

Definition at line 45 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::check_theta().

Here is the call graph for this function:



7.2.4.4 template<int Ndim> template<class IterT > static bool prior_hessian::AMHCopula< Ndim >::check_params_iter (IterT & *params*) [inline], [static]

Definition at line 49 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::check_theta().

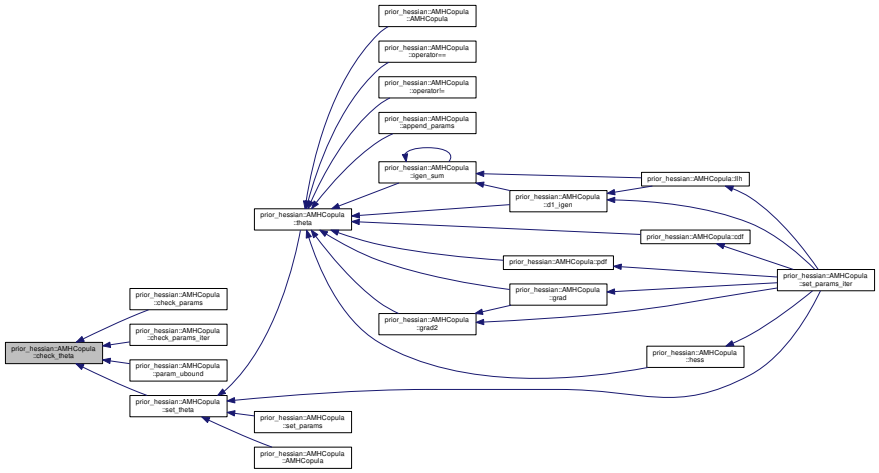
Here is the call graph for this function:

7.2.4.5 template<int Ndim> bool prior_hessian::AMHCopula< Ndim >::check_theta (double *val*) [static]

Definition at line 163 of file AMHCopula.h.

Referenced by prior_hessian::AMHCopula< Ndim >::check_params(), prior_hessian::AMHCopula< Ndim >::check_params_iter(), prior_hessian::AMHCopula< Ndim >::param_ubound(), and prior_hessian::AMHCopula< Ndim >::set_theta().

Here is the caller graph for this function:



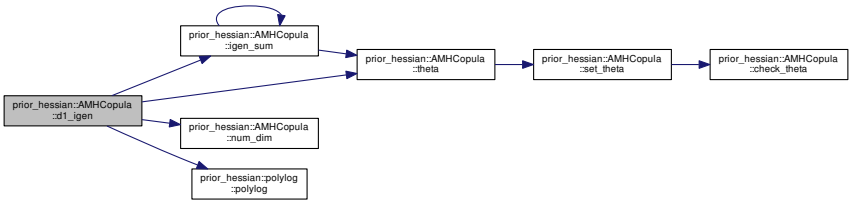
7.2.4.6 template<int Ndim> double prior_hessian::AMHCopula< Ndim >::d1_igen (double *u*) const

Definition at line 284 of file AMHCopula.h.

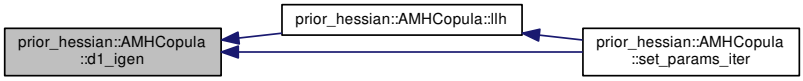
References prior_hessian::AMHCopula< Ndim >::igen_sum(), prior_hessian::AMHCopula< Ndim >::num_dim(), prior_hessian::polylog::polylog(), and prior_hessian::AMHCopula< Ndim >::theta().

Referenced by prior_hessian::AMHCopula< Ndim >::llh(), and prior_hessian::AMHCopula< Ndim >::set_params_↩ iter().

Here is the call graph for this function:



Here is the caller graph for this function:



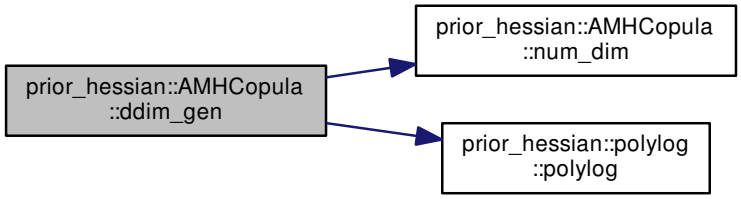
7.2.4.7 template<int Ndim> double prior_hessian::AMHCopula< Ndim >::ddim_gen (double *t*) const

Definition at line 270 of file AMHCopula.h.

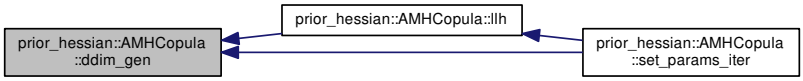
References prior_hessian::AMHCopula< Ndim >::num_dim(), and prior_hessian::polylog::polylog().

Referenced by prior_hessian::AMHCopula< Ndim >::llh(), and prior_hessian::AMHCopula< Ndim >::set_params_↩ iter().

Here is the call graph for this function:



Here is the caller graph for this function:

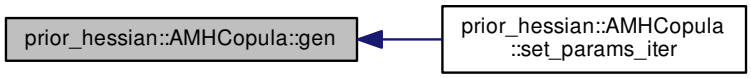


7.2.4.8 template<int Ndim> double prior_hessian::AMHCopula< Ndim >::gen (double *t*) const

Definition at line 265 of file AMHCopula.h.

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the caller graph for this function:



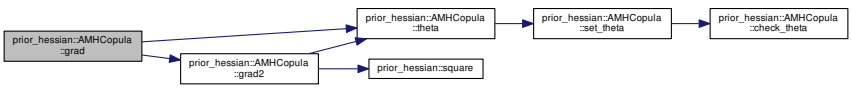
7.2.4.9 template<int Ndim> template<class Vec > AMHCopula< Ndim >::NdimVecT prior_hessian::AMHCopula< Ndim >::grad (const Vec & *u*) const

Definition at line 220 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::grad2(), and prior_hessian::AMHCopula< Ndim >::theta().

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:

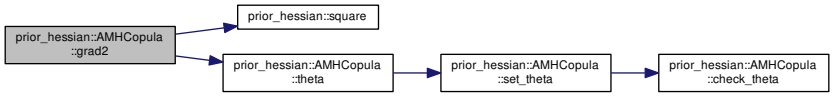
7.2.4.10 template<int Ndim> template<class Vec > AMHCopula< Ndim >::NdimVecT prior_hessian::AMHCopula< Ndim >::grad2 (const Vec & *u*) const

Definition at line 234 of file AMHCopula.h.

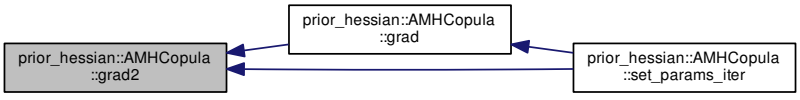
References prior_hessian::square(), and prior_hessian::AMHCopula< Ndim >::theta().

Referenced by prior_hessian::AMHCopula< Ndim >::grad(), and prior_hessian::AMHCopula< Ndim >::set_params↩
_iter().

Here is the call graph for this function:



Here is the caller graph for this function:



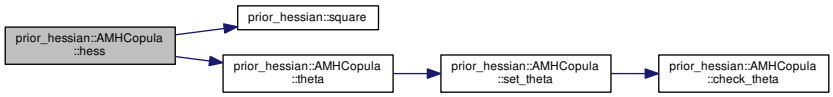
7.2.4.11 template<int Ndim> template<class Vec > AMHCopula< Ndim >::NdimMatT prior_hessian::AMHCopula< Ndim >::hess (const Vec & *u*) const

Definition at line 248 of file AMHCopula.h.

References prior_hessian::square(), and prior_hessian::AMHCopula< Ndim >::theta().

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:

7.2.4.12 template<int Ndim> double prior_hessian::AMHCopula< Ndim >::igen (double *u*) const

Definition at line 279 of file AMHCopula.h.

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the caller graph for this function:

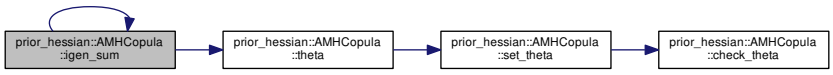
7.2.4.13 template<int Ndim> template<class Vec > double prior_hessian::AMHCopula< Ndim >::igen_sum (const Vec & *u*) [inline]

Definition at line 97 of file AMHCopula.h.

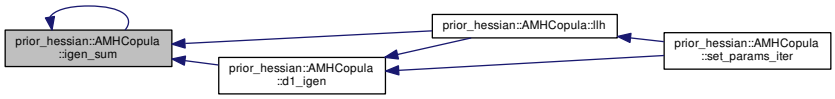
References prior_hessian::AMHCopula< Ndim >::igen_sum(), and prior_hessian::AMHCopula< Ndim >::theta().

Referenced by prior_hessian::AMHCopula< Ndim >::d1_igen(), prior_hessian::AMHCopula< Ndim >::igen_sum(), and prior_hessian::AMHCopula< Ndim >::llh().

Here is the call graph for this function:



Here is the caller graph for this function:



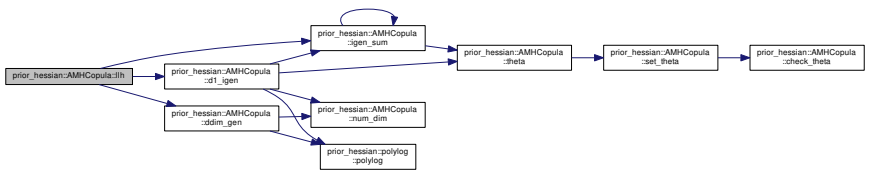
7.2.4.14 template<int Ndim> template<class Vec > double prior_hessian::AMHCopula< Ndim >::llh (const Vec & *u*) const

Definition at line 210 of file AMHCopula.h.

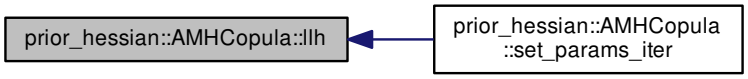
References prior_hessian::AMHCopula< Ndim >::d1_igen(), prior_hessian::AMHCopula< Ndim >::ddim_gen(), and prior_hessian::AMHCopula< Ndim >::igen_sum().

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:

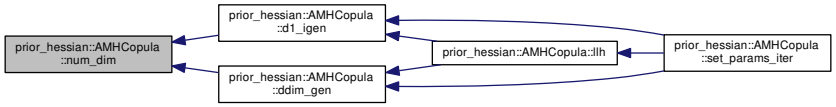


7.2.4.15 template<int Ndim> static constexpr IdxT prior_hessian::AMHCopula< Ndim >::num_dim ( ) [inline], [static]

Definition at line 30 of file AMHCopula.h.

Referenced by prior_hessian::AMHCopula< Ndim >::d1_igen(), and prior_hessian::AMHCopula< Ndim >::ddim_↩ gen().

Here is the caller graph for this function:



7.2.4.16 template<int Ndim> static constexpr IdxT prior_hessian::AMHCopula< Ndim >::num_params ( ) [inline], [static]

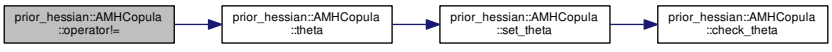
Definition at line 29 of file AMHCopula.h.

7.2.4.17 template<int Ndim> bool prior_hessian::AMHCopula< Ndim >::operator!= (const AMHCopula< Ndim > & *o*) const [inline]

Definition at line 42 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::theta().

Here is the call graph for this function:

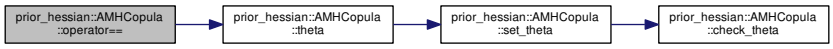


7.2.4.18 template<int Ndim> bool prior_hessian::AMHCopula< Ndim >::operator== (const AMHCopula< Ndim > & *o*) const [inline]

Definition at line 41 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::theta().

Here is the call graph for this function:



7.2.4.19 template<int Ndim> static double prior_hessian::AMHCopula< Ndim >::param_lbound ( ) [inline], [static]

Definition at line 31 of file AMHCopula.h.

7.2.4.20 template<int Ndim> static const StringVecT& prior_hessian::AMHCopula< Ndim >::param_names ( ) [inline], [static]

Definition at line 28 of file AMHCopula.h.

7.2.4.21 template<int Ndim> static double prior_hessian::AMHCopula< Ndim >::param_ubound ( ) [inline], [static]

Definition at line 32 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::check_theta().

Here is the call graph for this function:

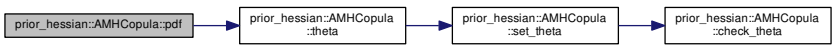
7.2.4.22 template<int Ndim> template<class Vec > double prior_hessian::AMHCopula< Ndim >::pdf (const Vec & *u*) const

Definition at line 202 of file AMHCopula.h.

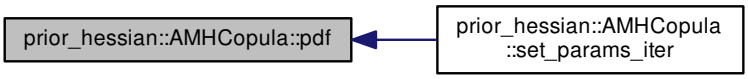
References prior_hessian::AMHCopula< Ndim >::theta().

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:



7.2.4.23 template<int Ndim> template<class Vec > double prior_hessian::AMHCopula< Ndim >::rllh (const Vec & *u*) const

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the caller graph for this function:

7.2.4.24 template<int Ndim> double prior_hessian::AMHCopula< Ndim >::rllh_const ( ) const

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the caller graph for this function:

7.2.4.25 template<int Ndim> template<class Vec > static void prior_hessian::AMHCopula< Ndim >::rllh_d2theta_accumulate (double *theta,* const Vec & *u,* double & *rllh,* double & *dtheta,* double & *d2theta*) [static]

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the caller graph for this function:

7.2.4.26 template<int Ndim> template<class Vec > static void prior_hessian::AMHCopula< Ndim >::rllh_dtheta_accumulate (double *theta,* const Vec & *u,* double & *rllh,* double & *dtheta*) [static]

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the caller graph for this function:

7.2.4.27 template<int Ndim> template<class Vec , class Vec2 > void prior_hessian::AMHCopula< Ndim >::rllh_grad_accumulate (const Vec & *u,* double & *rllh,* Vec2 & *grad*) const

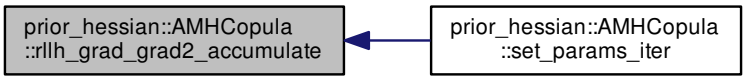
Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the caller graph for this function:

7.2.4.28 template<int Ndim> template<class Vec , class Vec2 > void prior_hessian::AMHCopula< Ndim >::rllh_grad_grad2_accumulate (const Vec & *u,* double & *rllh,* Vec2 & *grad,* Vec2 & *grad2*) const

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

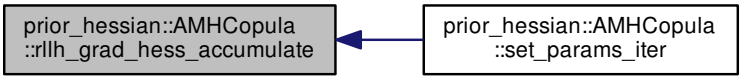
Here is the caller graph for this function:



7.2.4.29 template<int Ndim> template<class Vec , class Vec2 , class Mat > void prior_hessian::AMHCopula< Ndim >::rllh_grad_hess_accumulate (const Vec & *u,* double & *rllh,* Vec2 & *grad,* Mat & *hess*) const

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

Here is the caller graph for this function:



7.2.4.30 template<int Ndim> template<class RngT > AMHCopula< Ndim >::NdimVecT prior_hessian::AMHCopula< Ndim >::sample (RngT & *rng*) const

Definition at line 596 of file AMHCopula.h.

Referenced by prior_hessian::AMHCopula< Ndim >::set_params_iter().

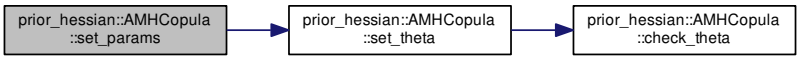
Here is the caller graph for this function:

7.2.4.31 template<int Ndim> template<class Vec > void prior_hessian::AMHCopula< Ndim >::set_params (const Vec & *params*) [inline]

Definition at line 47 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::set_theta().

Here is the call graph for this function:

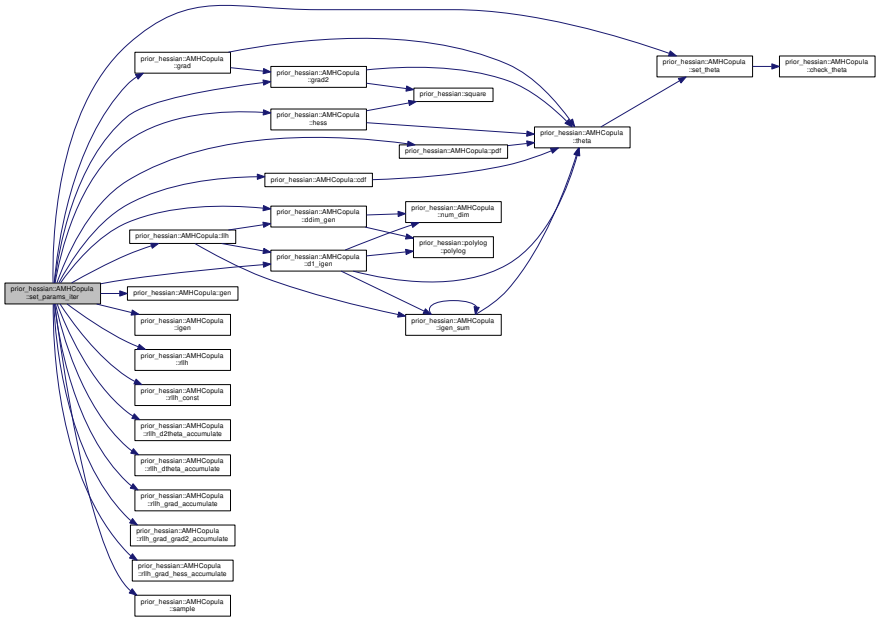


7.2.4.32 template<int Ndim> template<class IterT > void prior_hessian::AMHCopula< Ndim >::set_params_iter (IterT & *params*) [inline]

Definition at line 53 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::cdf(), prior_hessian::AMHCopula< Ndim >::d1_igen(), prior_hessian::AMHCopula< Ndim >::ddim_gen(), prior_hessian::AMHCopula< Ndim >::gen(), prior_hessian::AMHCopula< Ndim >::grad(), prior_hessian::AMHCopula< Ndim >::grad2(), prior_hessian::AMHCopula< Ndim >::hess(), prior_hessian::AMHCopula< Ndim >::igen(), prior_hessian::AMHCopula< Ndim >::llh(), prior_hessian::AMHCopula< Ndim >::pdf(), prior_hessian::AMHCopula< Ndim >::rllh(), prior_hessian::AMHCopula< Ndim >::rllh_const(), prior_hessian::AMHCopula< Ndim >::rllh_d2theta_accumulate(), prior_hessian::AMHCopula< Ndim >::rllh_dtheta_accumulate(), prior_hessian::AMHCopula< Ndim >::rllh_grad_accumulate(), prior_hessian::AMHCopula< Ndim >::rllh_grad_grad2_accumulate(), prior_hessian::AMHCopula< Ndim >::rllh_grad_hess_accumulate(), prior_hessian::AMHCopula< Ndim >::sample(), and prior_hessian::AMHCopula< Ndim >::set_theta().

Here is the call graph for this function:



7.2.4.33 template<int Ndim> void prior_hessian::AMHCopula< Ndim >::set_theta (double *val*)

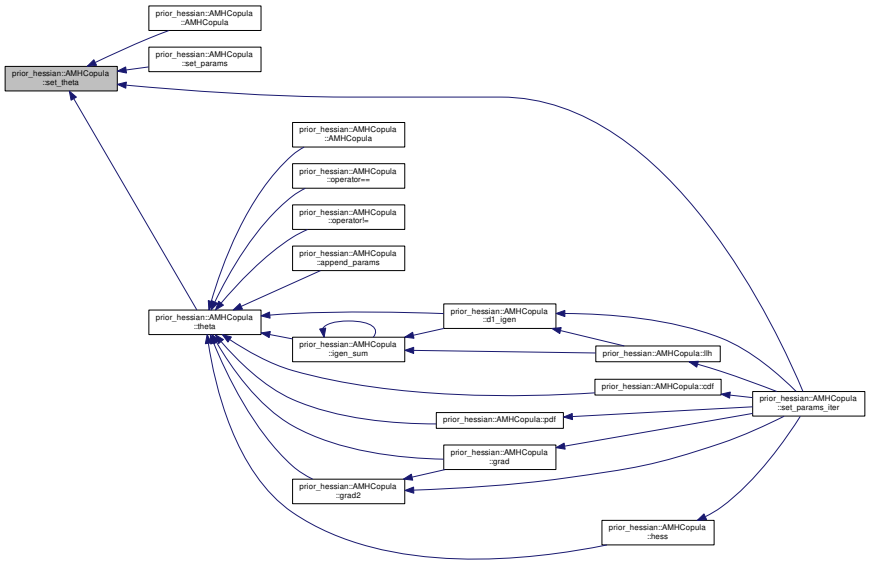
Definition at line 175 of file AMHCopula.h.

References prior_hessian::AMHCopula< Ndim >::check_theta().

Referenced by prior_hessian::AMHCopula< Ndim >::AMHCopula(), prior_hessian::AMHCopula< Ndim >::set_params(), prior_hessian::AMHCopula< Ndim >::set_params_iter(), and prior_hessian::AMHCopula< Ndim >::theta().

Here is the call graph for this function:

Here is the caller graph for this function:



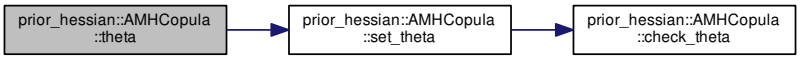
7.2.4.34 template<int Ndim> double prior_hessian::AMHCopula< Ndim >::theta ( ) const [inline]

Definition at line 38 of file AMHCopula.h.

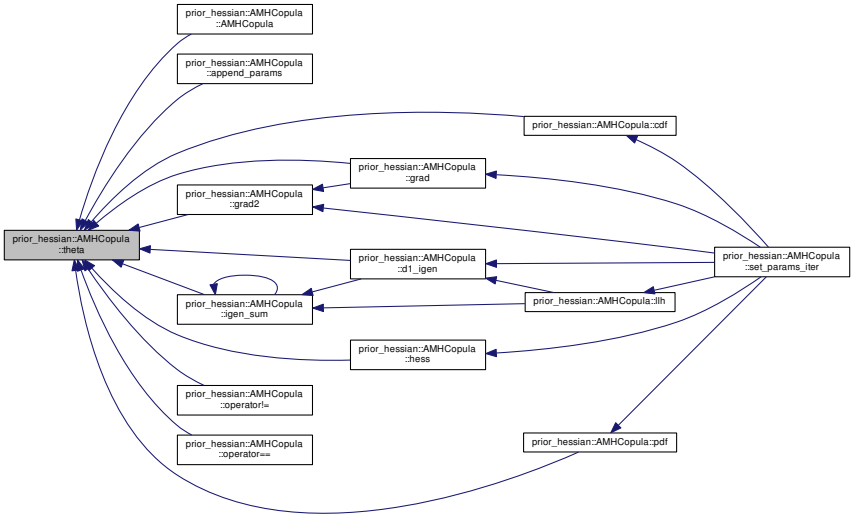
References prior_hessian::AMHCopula< Ndim >::set_theta().

Referenced by prior_hessian::AMHCopula< Ndim >::AMHCopula(), prior_hessian::AMHCopula< Ndim >::append_params(), prior_hessian::AMHCopula< Ndim >::cdf(), prior_hessian::AMHCopula< Ndim >::d1_igen(), prior_hessian::AMHCopula< Ndim >::grad(), prior_hessian::AMHCopula< Ndim >::grad2(), prior_hessian::AMHCopula< Ndim >::hess(), prior_hessian::AMHCopula< Ndim >::igen_sum(), prior_hessian::AMHCopula< Ndim >::operator!=(), prior_hessian::AMHCopula< Ndim >::operator==(), and prior_hessian::AMHCopula< Ndim >::pdf().

Here is the call graph for this function:



Here is the caller graph for this function:



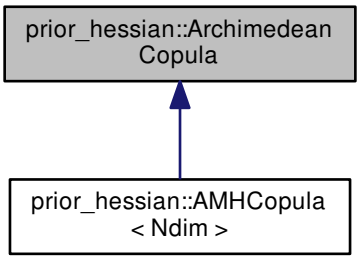
The documentation for this class was generated from the following file:

- AMHCopula.h

7.3 prior_hessian::ArchimedeanCopula Class Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ArchimedeanCopula.h>`

Inheritance diagram for prior_hessian::ArchimedeanCopula:



Classes

- struct D2_GenTerms
- struct D2_IGenTerms
- struct D2Theta_GenTerms
- struct D2Theta_IGenTerms
- struct D_GenTerms
- struct D_IGenTerms
- struct DTheta_GenTerms
- struct DTheta_IGenTerms

7.3.1 Detailed Description

Definition at line 11 of file ArchimedeanCopula.h.

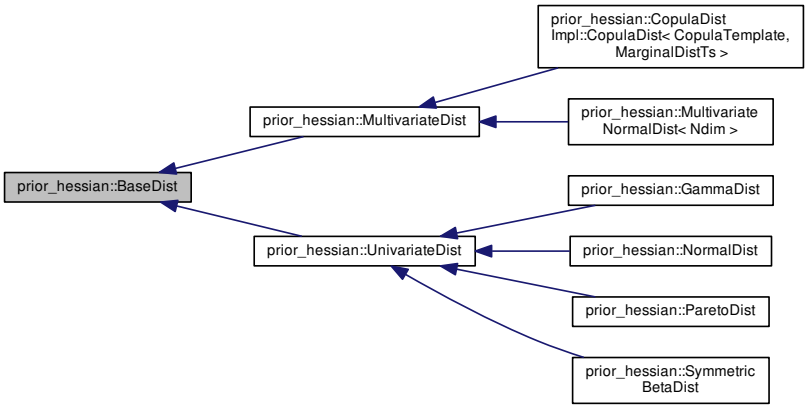
The documentation for this class was generated from the following file:

- ArchimedeanCopula.h

7.4 prior_hessian::BaseDist Class Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/BaseDist.h>`

Inheritance diagram for prior_hessian::BaseDist:



7.4.1 Detailed Description

Definition at line 11 of file BaseDist.h.

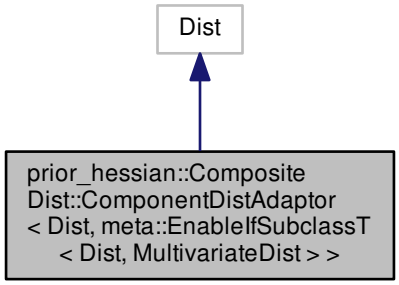
The documentation for this class was generated from the following file:

- BaseDist.h

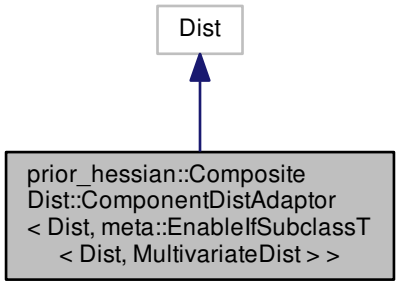
7.5 prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/CompositeDist.h>`

Inheritance diagram for prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >:



Collaboration diagram for prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >:



Public Types

- using ComponentDistT = Dist

Public Member Functions

- ComponentDistAdaptor ()
- ComponentDistAdaptor (Dist &&dist)
- ComponentDistAdaptor (const Dist &dist)
- template<class IterT >
  void append_lbound (IterT &v) const
- template<class IterT >
  void append_ubound (IterT &v) const
- template<class IterT >
  void append_global_lbound (IterT &v) const
- template<class IterT >
  void append_global_ubound (IterT &v) const
- template<class IterT >
  void set_lbound_from_iter (IterT &v)
- template<class IterT >
  void set_ubound_from_iter (IterT &v)
- template<class IterT >
  void set_bounds_from_iter (IterT &lb_iter, IterT &ub_iter)
- template<class IterT >
  void append_params (IterT &v) const
- template<class IterT >
  void append_params_lbound (IterT &v) const
- template<class IterT >
  void append_params_ubound (IterT &v) const
- template<class IterT >
  void append_param_names (IterT &v) const
- template<class IterT >
  double cdf_from_iter (IterT &u) const
- template<class IterT >
  double pdf_from_iter (IterT &u) const
- template<class IterT >
  double llh_from_iter (IterT &u) const
- template<class IterT >
  double rllh_from_iter (IterT &u) const
- void grad_accumulate_idx (const VecT &u, VecT &g, IdxT &k) const
- void grad2_accumulate_idx (const VecT &u, VecT &g2, IdxT &k) const
- void hess_accumulate_idx (const VecT &u, MatT &h, IdxT &k) const
- void grad_grad2_accumulate_idx (const VecT &u, VecT &g, VecT &g2, IdxT &k) const
- void grad_hess_accumulate_idx (const VecT &u, VecT &g, MatT &h, IdxT &k) const
- template<class RngT , class IterT >
  void append_sample (RngT &rng, IterT &v) const

7.5.1 Detailed Description

template<class Dist>
class prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >

Definition at line 696 of file CompositeDist.h.

7.5.2 Member Typedef Documentation

7.5.2.1 template<class Dist > using prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::ComponentDistT = Dist

Definition at line 698 of file CompositeDist.h.

7.5.3 Constructor & Destructor Documentation

7.5.3.1 template<class Dist > prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::ComponentDistAdaptor ( ) [inline]

Definition at line 699 of file CompositeDist.h.

7.5.3.2 template<class Dist > prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::ComponentDistAdaptor (Dist && *dist*) [inline], [explicit]

Definition at line 700 of file CompositeDist.h.

7.5.3.3 template<class Dist > prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::ComponentDistAdaptor (const Dist & *dist*) [inline], [explicit]

Definition at line 701 of file CompositeDist.h.

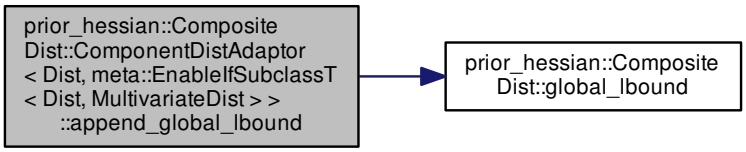
7.5.4 Member Function Documentation

7.5.4.1 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_global_lbound (IterT & *v*) const [inline]

Definition at line 709 of file CompositeDist.h.

References prior_hessian::CompositeDist::global_lbound().

Here is the call graph for this function:

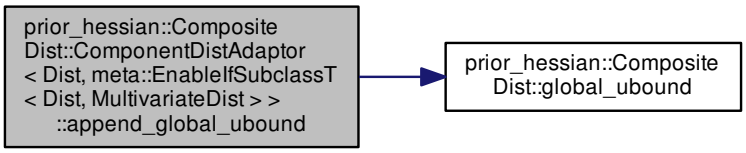


7.5.4.2 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_global_ubound (IterT & *v*) const [inline]

Definition at line 712 of file CompositeDist.h.

References prior_hessian::CompositeDist::global_ubound().

Here is the call graph for this function:

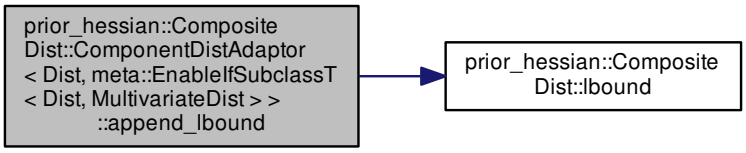


7.5.4.3 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_lbound (IterT & *v*) const [inline]

Definition at line 703 of file CompositeDist.h.

References prior_hessian::CompositeDist::lbound().

Here is the call graph for this function:

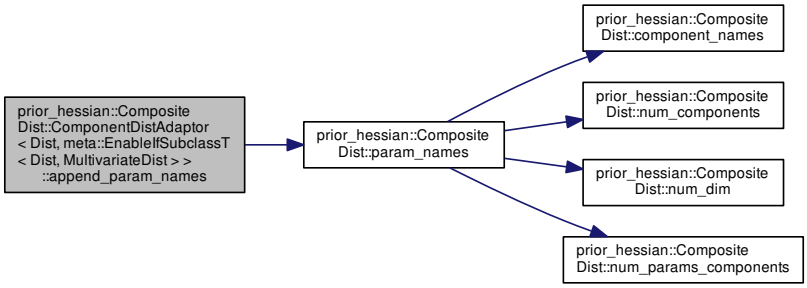


7.5.4.4 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_param_names (IterT & *v*) const [inline]

Definition at line 750 of file CompositeDist.h.

References prior_hessian::CompositeDist::param_names().

Here is the call graph for this function:

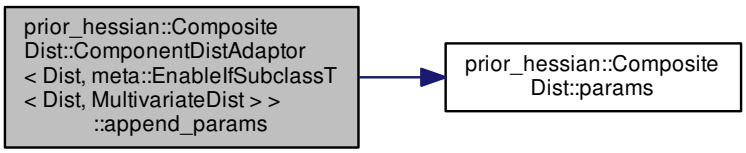


7.5.4.5 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_params (IterT & *v*) const [inline]

Definition at line 741 of file CompositeDist.h.

References prior_hessian::CompositeDist::params().

Here is the call graph for this function:



7.5.4.6 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_params_lbound (IterT & *v*) const [inline]

Definition at line 744 of file CompositeDist.h.

7.5.4.7 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_params_ubound (IterT & *v*) const [inline]

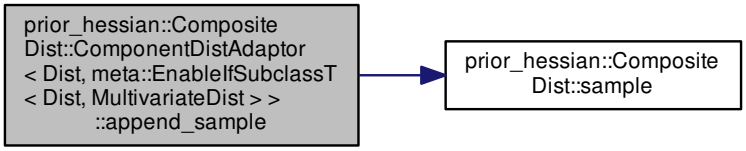
Definition at line 747 of file CompositeDist.h.

7.5.4.8 template<class Dist > template<class RngT , class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_sample (RngT & *rng,* IterT & *v*) const [inline]

Definition at line 827 of file CompositeDist.h.

References prior_hessian::CompositeDist::sample().

Here is the call graph for this function:

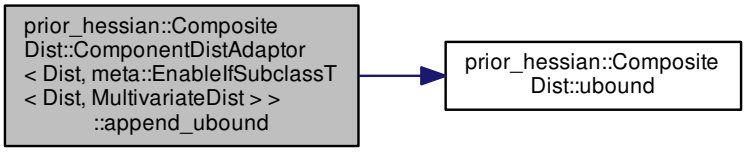


7.5.4.9 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_ubound (IterT & *v*) const [inline]

Definition at line 706 of file CompositeDist.h.

References prior_hessian::CompositeDist::ubound().

Here is the call graph for this function:

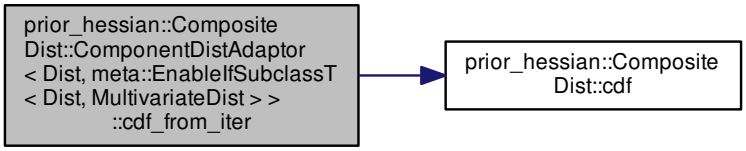


7.5.4.10 template<class Dist > template<class IterT > double prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::cdf_from_iter (IterT & *u*) const [inline]

Definition at line 753 of file CompositeDist.h.

References prior_hessian::CompositeDist::cdf().

Here is the call graph for this function:

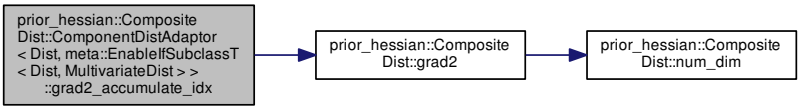


7.5.4.11 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad2_accumulate_idx (const VecT & *u,* VecT & *g2,* IdxT & *k*) const [inline]

Definition at line 792 of file CompositeDist.h.

References prior_hessian::CompositeDist::grad2().

Here is the call graph for this function:

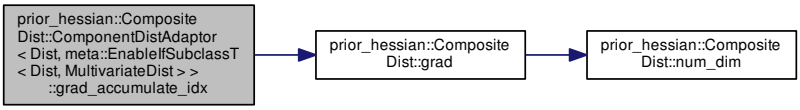


7.5.4.12 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad_accumulate_idx (const VecT & *u,* VecT & *g,* IdxT & *k*) const [inline]

Definition at line 785 of file CompositeDist.h.

References prior_hessian::CompositeDist::grad().

Here is the call graph for this function:

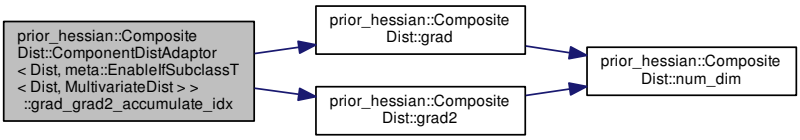


7.5.4.13 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad_grad2_accumulate_idx (const VecT & *u*, VecT & *g*, VecT & *g2*, IdxT & *k*) const [inline]

Definition at line 807 of file CompositeDist.h.

References prior_hessian::CompositeDist::grad(), and prior_hessian::CompositeDist::grad2().

Here is the call graph for this function:

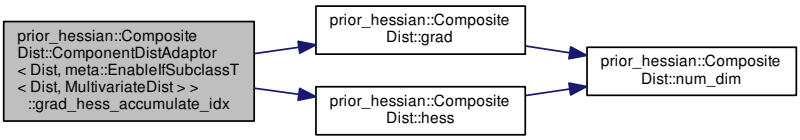


7.5.4.14 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad_hess_accumulate_idx (const VecT & *u*, VecT & *g*, MatT & *h*, IdxT & *k*) const [inline]

Definition at line 816 of file CompositeDist.h.

References prior_hessian::CompositeDist::grad(), and prior_hessian::CompositeDist::hess().

Here is the call graph for this function:

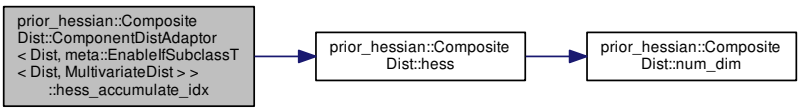


7.5.4.15 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::hess_accumulate_idx (const VecT & *u*, MatT & *h*, IdxT & *k*) const [inline]

Definition at line 799 of file CompositeDist.h.

References prior_hessian::CompositeDist::hess().

Here is the call graph for this function:

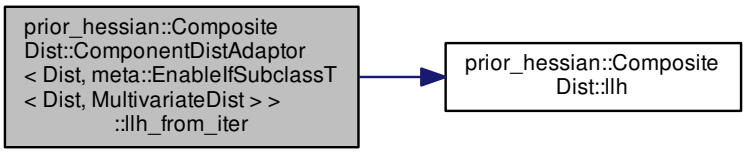


7.5.4.16 template<class Dist > template<class IterT > double prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::llh_from_iter (IterT & *u*) const [inline]

Definition at line 769 of file CompositeDist.h.

References prior_hessian::CompositeDist::llh().

Here is the call graph for this function:

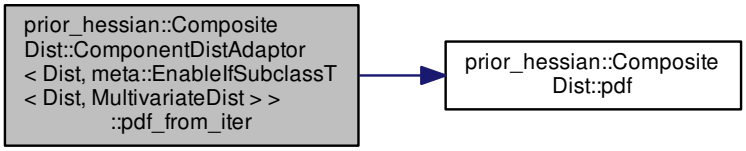


7.5.4.17 template<class Dist > template<class IterT > double prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::pdf_from_iter (IterT & *u*) const [inline]

Definition at line 761 of file CompositeDist.h.

References prior_hessian::CompositeDist::pdf().

Here is the call graph for this function:

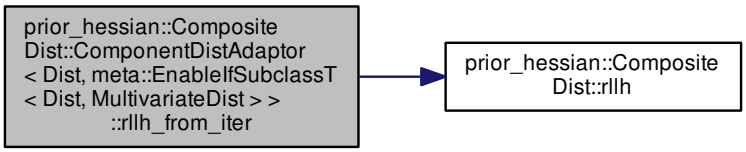


7.5.4.18 template<class Dist > template<class IterT > double prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::rllh_from_iter (IterT & *u*) const [inline]

Definition at line 777 of file CompositeDist.h.

References prior_hessian::CompositeDist::rllh().

Here is the call graph for this function:

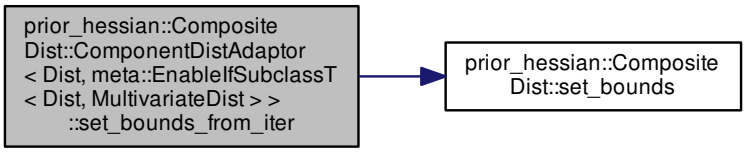


7.5.4.19 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::set_bounds_from_iter (IterT & *lb_iter,* IterT & *ub_iter*) [inline]

Definition at line 731 of file CompositeDist.h.

References prior_hessian::CompositeDist::set_bounds().

Here is the call graph for this function:

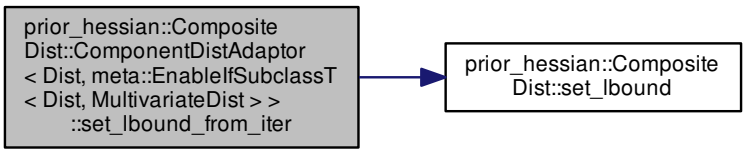


7.5.4.20 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::set_lbound_from_iter (IterT & *v*) [inline]

Definition at line 715 of file CompositeDist.h.

References prior_hessian::CompositeDist::set_lbound().

Here is the call graph for this function:

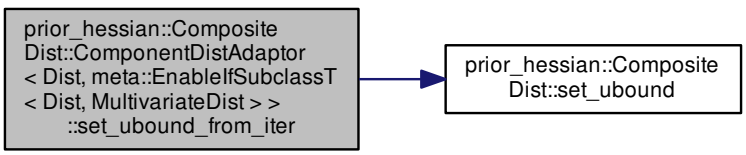


7.5.4.21 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::set_ubound_from_iter (IterT & *v*) [inline]

Definition at line 723 of file CompositeDist.h.

References prior_hessian::CompositeDist::set_ubound().

Here is the call graph for this function:



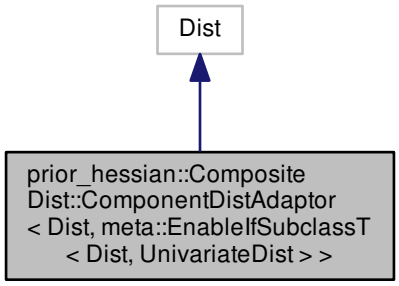
The documentation for this class was generated from the following file:

- CompositeDist.h

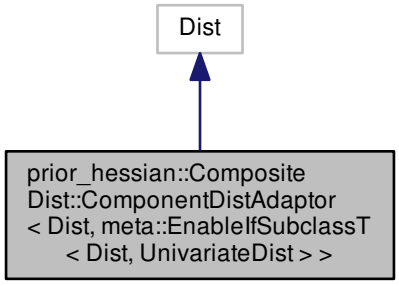
7.6 prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/CompositeDist.h>`

Inheritance diagram for prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >:



Collaboration diagram for prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >:



Public Types

- using ComponentDistT = Dist

Public Member Functions

- ComponentDistAdaptor ()
- ComponentDistAdaptor (Dist &&dist)
- ComponentDistAdaptor (const Dist &dist)
- template<class IterT >
  void append_lbound (IterT &v) const
- template<class IterT >
  void append_ubound (IterT &v) const
- template<class IterT >
  void append_global_lbound (IterT &v) const
- template<class IterT >
  void append_global_ubound (IterT &v) const
- template<class IterT >
  void set_lbound_from_iter (IterT &lbounds)
- template<class IterT >
  void set_ubound_from_iter (IterT &ubounds)
- template<class IterT >
  void set_bounds_from_iter (IterT &lbounds, IterT &ubounds)
- template<class IterT >
  void append_params (IterT &v) const
- template<class IterT >
  void append_params_lbound (IterT &v) const
- template<class IterT >
  void append_params_ubound (IterT &v) const
- template<class IterT >
  void append_param_names (IterT &v) const
- template<class IterT >
  double cdf_from_iter (IterT &u) const
- template<class IterT >
  double pdf_from_iter (IterT &u) const
- template<class IterT >
  double llh_from_iter (IterT &u) const
- template<class IterT >
  double rllh_from_iter (IterT &u) const
- void grad_accumulate_idx (const VecT &u, VecT &g, IdxT &k) const
- void grad2_accumulate_idx (const VecT &u, VecT &g2, IdxT &k) const
- void hess_accumulate_idx (const VecT &u, MatT &h, IdxT &k) const
- void grad_grad2_accumulate_idx (const VecT &u, VecT &g, VecT &g2, IdxT &k) const
- void grad_hess_accumulate_idx (const VecT &u, VecT &g, MatT &h, IdxT &k) const
- template<class RngT , class IterT >
  void append_sample (RngT &rng, IterT &iter) const

7.6.1 Detailed Description

template<class Dist>
class prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >

Definition at line 626 of file CompositeDist.h.

7.6.2 Member Typedef Documentation

7.6.2.1 template<class Dist > using prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::ComponentDistT = Dist

Definition at line 628 of file CompositeDist.h.

7.6.3 Constructor & Destructor Documentation

7.6.3.1 template<class Dist > prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::ComponentDistAdaptor ( ) [inline]

Definition at line 629 of file CompositeDist.h.

7.6.3.2 template<class Dist > prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::ComponentDistAdaptor (Dist && *dist*) [inline], [explicit]

Definition at line 630 of file CompositeDist.h.

7.6.3.3 template<class Dist > prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::ComponentDistAdaptor (const Dist & *dist*) [inline], [explicit]

Definition at line 631 of file CompositeDist.h.

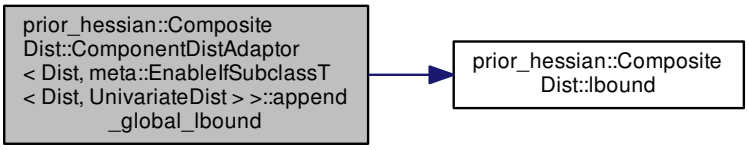
7.6.4 Member Function Documentation

7.6.4.1 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_global_lbound (IterT & *v*) const [inline]

Definition at line 635 of file CompositeDist.h.

References prior_hessian::CompositeDist::lbound().

Here is the call graph for this function:

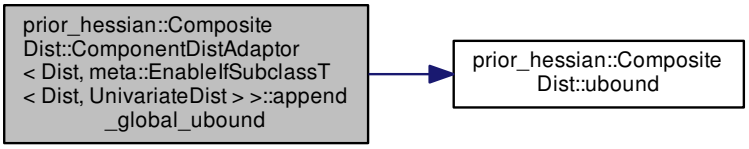


7.6.4.2 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_global_ubound (IterT & *v*) const [inline]

Definition at line 636 of file CompositeDist.h.

References prior_hessian::CompositeDist::ubound().

Here is the call graph for this function:

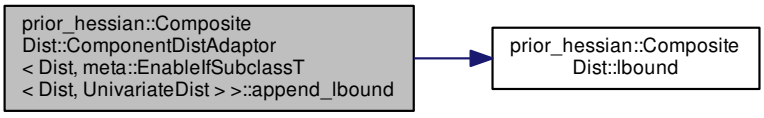


7.6.4.3 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_lbound (IterT & *v*) const [inline]

Definition at line 633 of file CompositeDist.h.

References prior_hessian::CompositeDist::lbound().

Here is the call graph for this function:

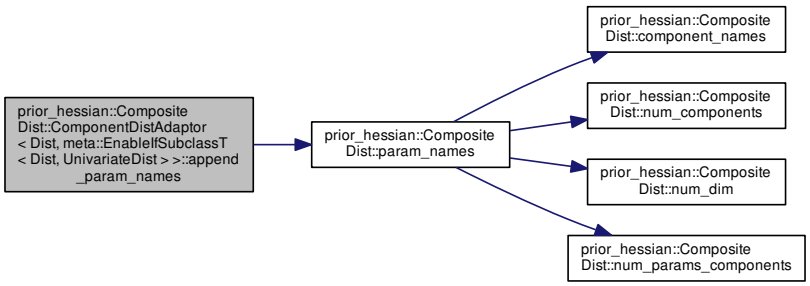


7.6.4.4 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_param_names (IterT & *v*) const [inline]

Definition at line 652 of file CompositeDist.h.

References prior_hessian::CompositeDist::param_names().

Here is the call graph for this function:

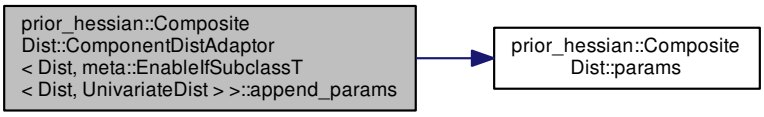


7.6.4.5 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_params (IterT & *v*) const [inline]

Definition at line 643 of file CompositeDist.h.

References prior_hessian::CompositeDist::params().

Here is the call graph for this function:



7.6.4.6 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_params_lbound (IterT & *v*) const [inline]

Definition at line 646 of file CompositeDist.h.

7.6.4.7 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_params_ubound (IterT & *v*) const [inline]

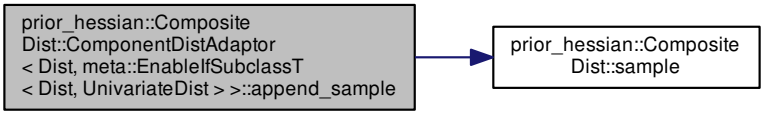
Definition at line 649 of file CompositeDist.h.

7.6.4.8 template<class Dist > template<class RngT , class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_sample (RngT & *rng,* IterT & *iter*) const [inline]

Definition at line 691 of file CompositeDist.h.

References prior_hessian::CompositeDist::sample().

Here is the call graph for this function:

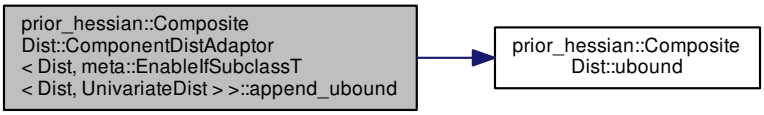


7.6.4.9 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_ubound (IterT & *v*) const [inline]

Definition at line 634 of file CompositeDist.h.

References prior_hessian::CompositeDist::ubound().

Here is the call graph for this function:

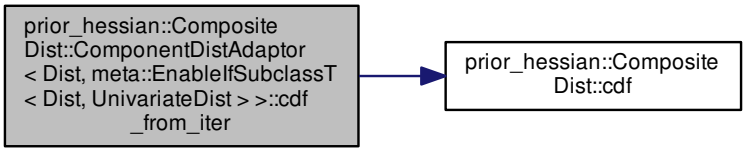


7.6.4.10 template<class Dist > template<class IterT > double prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::cdf_from_iter (IterT & *u*) const [inline]

Definition at line 655 of file CompositeDist.h.

References prior_hessian::CompositeDist::cdf().

Here is the call graph for this function:

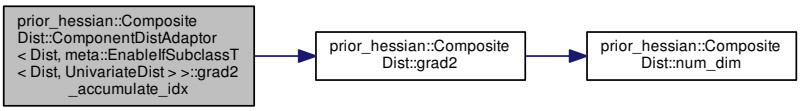


7.6.4.11 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::grad2_accumulate_idx (const VecT & *u,* VecT & *g2,* IdxT & *k*) const [inline]

Definition at line 666 of file CompositeDist.h.

References prior_hessian::CompositeDist::grad2().

Here is the call graph for this function:

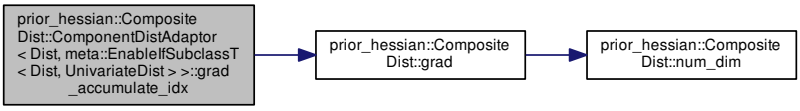


7.6.4.12 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::grad_accumulate_idx (const VecT & *u,* VecT & *g,* IdxT & *k*) const [inline]

Definition at line 660 of file CompositeDist.h.

References prior_hessian::CompositeDist::grad().

Here is the call graph for this function:

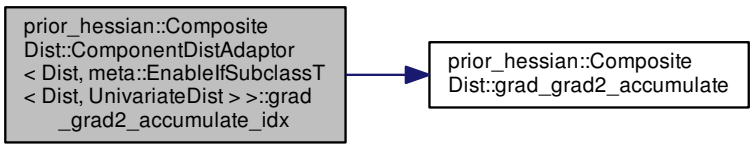


7.6.4.13 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::grad_grad2_accumulate_idx (const VecT & *u*, VecT & *g*, VecT & *g2*, IdxT & *k*) const [inline]

Definition at line 678 of file CompositeDist.h.

References prior_hessian::CompositeDist::grad_grad2_accumulate().

Here is the call graph for this function:

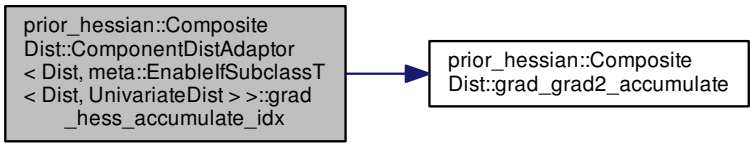


7.6.4.14 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::grad_hess_accumulate_idx (const VecT & *u*, VecT & *g*, MatT & *h*, IdxT & *k*) const [inline]

Definition at line 684 of file CompositeDist.h.

References prior_hessian::CompositeDist::grad_grad2_accumulate().

Here is the call graph for this function:

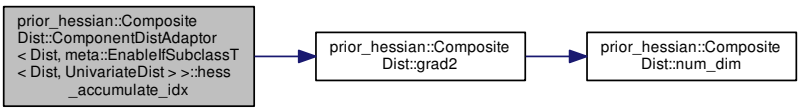


7.6.4.15 template<class Dist > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::hess_accumulate_idx (const VecT & *u,* MatT & *h,* IdxT & *k*) const [inline]

Definition at line 672 of file CompositeDist.h.

References prior_hessian::CompositeDist::grad2().

Here is the call graph for this function:

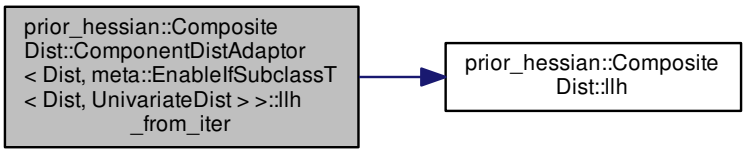


7.6.4.16 template<class Dist > template<class IterT > double prior_hessian::CompositeDist::ComponentDist↩ Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::llh_from_iter (IterT & *u*) const [inline]

Definition at line 657 of file CompositeDist.h.

References prior_hessian::CompositeDist::llh().

Here is the call graph for this function:

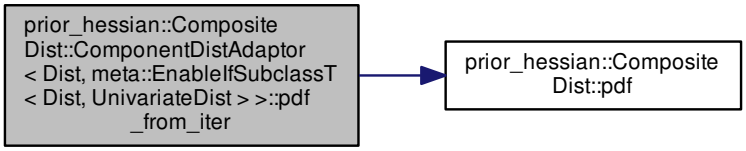


7.6.4.17 template<class Dist > template<class IterT > double prior_hessian::CompositeDist::ComponentDist↩ Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::pdf_from_iter (IterT & *u*) const [inline]

Definition at line 656 of file CompositeDist.h.

References prior_hessian::CompositeDist::pdf().

Here is the call graph for this function:

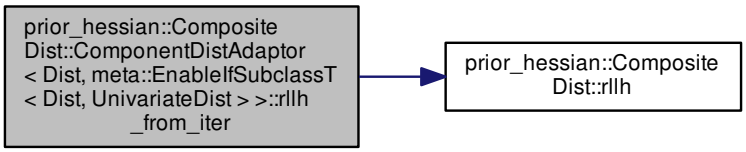


7.6.4.18 template<class Dist > template<class IterT > double prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::rllh_from_iter (IterT & *u*) const [inline]

Definition at line 658 of file CompositeDist.h.

References prior_hessian::CompositeDist::rllh().

Here is the call graph for this function:

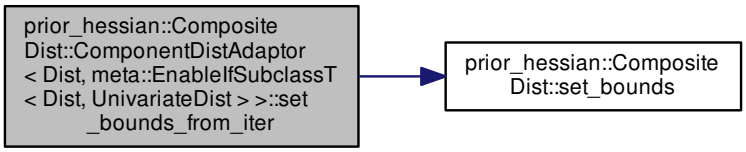


7.6.4.19 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::set_bounds_from_iter (IterT & *lbounds,* IterT & *ubounds*) [inline]

Definition at line 640 of file CompositeDist.h.

References prior_hessian::CompositeDist::set_bounds().

Here is the call graph for this function:

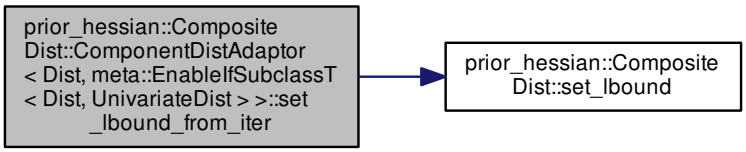


7.6.4.20 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::set_lbound_from_iter (IterT & *lbounds*) [inline]

Definition at line 637 of file CompositeDist.h.

References prior_hessian::CompositeDist::set_lbound().

Here is the call graph for this function:

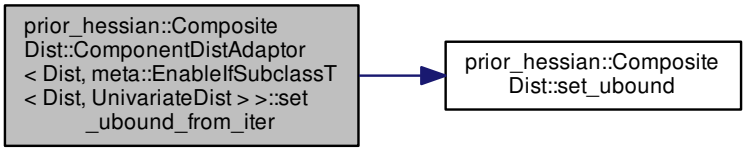


7.6.4.21 template<class Dist > template<class IterT > void prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::set_ubound_from_iter (IterT & *ubounds*) [inline]

Definition at line 638 of file CompositeDist.h.

References prior_hessian::CompositeDist::set_ubound().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- CompositeDist.h

7.7 prior_hessian::CompositeDist Class Reference

A probability distribution made of independent component distributions composing groups of 1 or more variables.

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ CompositeDist.h>`

Classes

- class ComponentDistAdaptor
- class ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >
- class ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >

Public Types

- using AnyRngT = any_rng::AnyRng< std::size_t >
- template<class DistT >
  using ComponentDistT = ComponentDistAdaptor< BoundsAdaptedDistT< std::remove_reference_t< DistT >>>

Public Member Functions

- CompositeDist ()
- template<class... Ts, meta::ConstructableIfAllAreNotTupleAndAreNotT< CompositeDist, Ts... > = true>
  CompositeDist (Ts &&...dists)

  Construct from a variadic list of subclasses of UnivariateDist's or MulitvariateDist's.
- template<class... Ts>
  CompositeDist (std::tuple< Ts... > &&dist_tuple)

  Construct from a rvalue tuple of subclasses of UnivariateDist's or MulitvariateDist's.
- template<class... Ts>
  CompositeDist (const std::tuple< Ts... > &dist_tuple)

  Construct from a lvalue tuple of subclasses of UnivariateDist's or MulitvariateDist's.
- void initialize ()
- void initialize (const std::tuple<> &)

  Initialize to the empty state.
- void initialize (std::tuple<> &&)

  Initialize of an empty lvalue tuple produces the empty state.
- template<class... Ts, typename = meta::EnableIfAllAreNotTupleT<Ts...>>
  void initialize (Ts &&...dists)

  Initialize of an empty rvalue tuple produces the empty state.
- template<class... Ts, typename = meta::EnableIfAllAreNotTupleT<Ts...>>
  void initialize (std::tuple< Ts... > &&dist_tuple)
- template<class... Ts, typename = meta::EnableIfNonEmpty<Ts...>>
  void initialize (const std::tuple< Ts... > &dist_tuple)
- CompositeDist (const CompositeDist &)
- CompositeDist & operator= (const CompositeDist &)
- CompositeDist (CompositeDist &&)
- CompositeDist & operator= (CompositeDist &&)
- void clear ()
- template<class... Ts>
  const std::tuple< Ts... > & get_dist_tuple () const
- bool is_empty () const
- operator bool () const
- IdxT num_components () const
- TypeInfoVecT component_types () const
- const StringVecT & component_names () const
- template<class StringVec >
  void set_component_names (StringVec &&names)
- bool operator== (const CompositeDist &o) const
- bool operator!= (const CompositeDist &o) const
- IdxT num_dim () const
- UVecT num_dim_components () const
- const StringVecT & dim_variables () const
- template<class StringVec >
  void set_dim_variables (StringVec &&vars)
- VecT lbound () const
- VecT ubound () const
- VecT global_lbound () const
- VecT global_ubound () const
- bool in_bounds (const VecT &u) const

- bool in_bounds_all (const MatT &u) const
- void set_lbound (const VecT &new_bound)
- void set_ubound (const VecT &new_bound)
- void set_bounds (const VecT &new_lbound, const VecT &new_ubound)
- IdxT num_params () const
- UVecT num_params_components () const
- VecT params () const
- void set_params (const VecT &new_params)
- bool check_params (const VecT &new_params) const
- VecT params_lbound () const
- VecT params_ubound () const
- std::vector< VecT > params_components () const
- const StringVecT & param_names () const
- template<class StringVec >
  void set_param_names (StringVec &&vars)
- bool has_dim_variable (const std::string &name) const
- IdxT get_dim_variable_index (const std::string &name) const
- void rename_dim_variable (const std::string &old_name, std::string new_name)
- bool has_param (const std::string &name) const
- double get_param_value (const std::string &name) const
- IdxT get_param_index (const std::string &name) const
- void set_param_value (const std::string &name, double value)
- void rename_param (const std::string &old_name, std::string new_name)
- double cdf (const VecT &u) const
- double pdf (const VecT &u) const
- double llh (const VecT &u) const
- double rllh (const VecT &u) const
- VecT grad (const VecT &u) const
- VecT grad2 (const VecT &u) const
- MatT hess (const VecT &u) const
- void grad_accumulate (const VecT &theta, VecT &grad) const
- void grad2_accumulate (const VecT &theta, VecT &grad2) const
- void hess_accumulate (const VecT &theta, MatT &hess) const
- void grad_grad2_accumulate (const VecT &theta, VecT &grad, VecT &grad2) const
- void grad_hess_accumulate (const VecT &theta, VecT &grad, MatT &hess) const
- VecT make_zero_grad () const
- MatT make_zero_hess () const
- VecT sample (AnyRngT &rng) const
- VecT sample (AnyRngT &&rng) const
- MatT sample (AnyRngT &rng, IdxT num_samples) const
- MatT sample (AnyRngT &&rng, IdxT num_samples) const
- template<class RngT >
  VecT sample (RngT &&rng) const
- template<class RngT >
  MatT sample (RngT &&rng, IdxT num_samples) const
- VecT llh_components (const VecT &u) const
- VecT rllh_components (const VecT &u) const

Static Public Member Functions

- template<class DistT >
  static meta::ReturnIfInstantiatedFromT< DistT, DistT, ComponentDistAdaptor > make_component_dist (DistT &&dist)
- template<class DistT >
  static meta::ReturnIfNotInstantiatedFromT< ComponentDistT< DistT >, DistT, ComponentDistAdaptor > make_component_dist (DistT &&dist)
- template<class... Ts>
  static std::tuple< ComponentDistT< Ts >... > make_component_dist_tuple (const std::tuple< Ts... > &dists)
- template<class... Ts, std::size_t... I>
  static std::tuple< ComponentDistT< Ts >... > make_component_dist_tuple (const std::tuple< Ts... > &dists, std::index_sequence< I... >)
- template<class... Ts>
  static std::tuple< ComponentDistT< Ts >... > make_component_dist_tuple (std::tuple< Ts... > &&dists)
- template<class... Ts, std::size_t... I>
  static std::tuple< ComponentDistT< Ts >... > make_component_dist_tuple (std::tuple< Ts... > &&dists, std::index_sequence< I... >)

7.7.1 Detailed Description

A probability distribution made of independent component distributions composing groups of 1 or more variables.

CompositeDist is a world unto itself.

class UnivariateDistInterface { static CONSTEXPR IdxT num_dim(); static CONSTEXPR IdxT num_params(); static const StringVecT param_names; double lbound() const; double ubound() const; void set_bounds(double lbound, double ubound); void set_lbound(double lbound); void set_ubound(double ubound); double get_param(int idx) const; void set_param(int idx, double val); double cdf(double x) const; double icdf(double u) const; double pdf(double x) const; double llh(double x) const; double rllh(double x) const; double grad(double x) const; double grad2(double x) const; void grad_grad2_accumulate(double x, double &g, double &g2) const; template< class RngT > double sample(RngT &rng) const; }

dim_variables and param_names are lazily computed. If they are not accessed, they are not created.

Definition at line 60 of file CompositeDist.h.

7.7.2 Class Documentation

7.7.2.1 class prior_hessian::CompositeDist::ComponentDistAdaptor

template<class DistT, typename Enable = void>
class prior_hessian::CompositeDist::ComponentDistAdaptor< DistT, Enable >

Definition at line 64 of file CompositeDist.h.

7.7.3 Member Typedef Documentation

7.7.3.1 using prior_hessian::CompositeDist::AnyRngT = any_rng::AnyRng<std::size_t>

Definition at line 67 of file CompositeDist.h.

7.7.3.2 template<class DistT > using prior_hessian::CompositeDist::ComponentDistT = ComponentDistAdaptor<BoundsAdaptedDistT<std::remove_reference_t<DistT>>>

Definition at line 844 of file CompositeDist.h.

7.7.4 Constructor & Destructor Documentation

7.7.4.1 prior_hessian::CompositeDist::CompositeDist ( )

Definition at line 13 of file CompositeDist.cpp.

Referenced by initialize().

Here is the caller graph for this function:

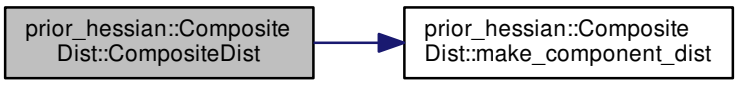
7.7.4.2 template<class... Ts, meta::ConstructableIfAllAreNotTupleAndAreNotT< CompositeDist, Ts... > = true> prior_hessian::CompositeDist::CompositeDist (Ts &&... *dists*) [inline], [explicit]

Construct from a variadic list of subclasses of UnivariateDist's or MulitvariateDist's.

Definition at line 73 of file CompositeDist.h.

References make_component_dist().

Here is the call graph for this function:



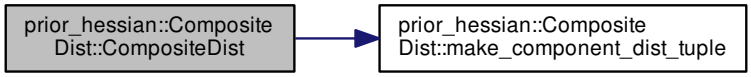
7.7.4.3 template<class... Ts> prior_hessian::CompositeDist::CompositeDist (std::tuple< Ts... > && *dist_tuple*) [inline], [explicit]

Construct from a rvalue tuple of subclasses of UnivariateDist's or MulitvariateDist's.

Definition at line 79 of file CompositeDist.h.

References make_component_dist_tuple().

Here is the call graph for this function:



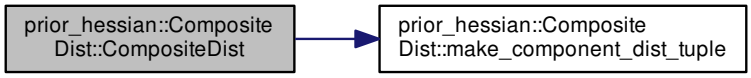
7.7.4.4 template<class... Ts> prior_hessian::CompositeDist::CompositeDist (const std::tuple< Ts... > & *dist_tuple*) [inline], [explicit]

Construct from a lvalue tuple of subclasses of UnivariateDist's or MulitvariateDist's.

Definition at line 85 of file CompositeDist.h.

References make_component_dist_tuple().

Here is the call graph for this function:



7.7.4.5 prior_hessian::CompositeDist::CompositeDist (const CompositeDist & *o*)

Definition at line 17 of file CompositeDist.cpp.

7.7.4.6 prior_hessian::CompositeDist::CompositeDist (CompositeDist && *o*)

Definition at line 28 of file CompositeDist.cpp.

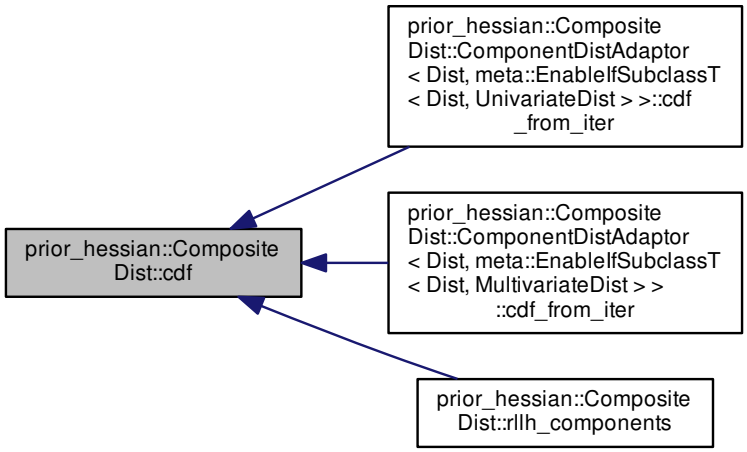
7.7.5 Member Function Documentation

7.7.5.1 double prior_hessian::CompositeDist::cdf (const VecT & *u*) const [inline]

Definition at line 186 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::cdf_from_iter(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIf↩SubclassT< Dist, MultivariateDist > >::cdf_from_iter(), and rllh_components().

Here is the caller graph for this function:



7.7.5.2 bool prior_hessian::CompositeDist::check_params (const VecT & *new_params*) const [inline]

Definition at line 163 of file CompositeDist.h.

Referenced by rllh_components().

Here is the caller graph for this function:

7.7.5.3 void prior_hessian::CompositeDist::clear ( )

Definition at line 38 of file CompositeDist.cpp.

Referenced by initialize().

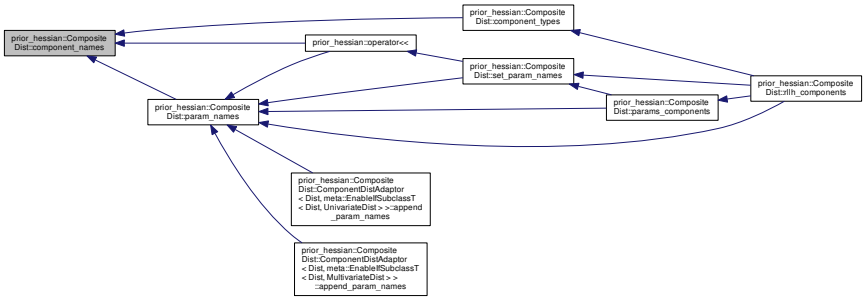
Here is the caller graph for this function:

7.7.5.4 const StringVecT & prior_hessian::CompositeDist::component_names ( ) const

Definition at line 108 of file CompositeDist.cpp.

Referenced by component_types(), prior_hessian::operator<<(), and param_names().

Here is the caller graph for this function:



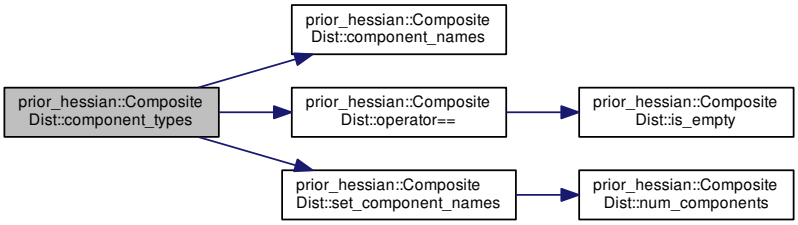
7.7.5.5 TypeInfoVecT prior_hessian::CompositeDist::component_types ( ) const [inline]

Definition at line 127 of file CompositeDist.h.

References component_names(), operator==(), and set_component_names().

Referenced by rllh_components().

Here is the call graph for this function:



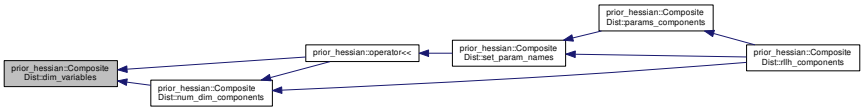
Here is the caller graph for this function:

7.7.5.6 const StringVecT & prior_hessian::CompositeDist::dim_variables ( ) const

Definition at line 114 of file CompositeDist.cpp.

Referenced by num_dim_components(), and prior_hessian::operator<<().

Here is the caller graph for this function:

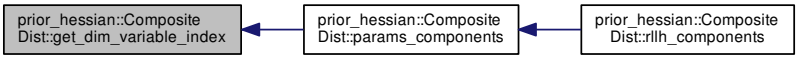


7.7.5.7 IdxT prior_hessian::CompositeDist::get_dim_variable_index (const std::string & *name*) const

Definition at line 175 of file CompositeDist.cpp.

Referenced by params_components().

Here is the caller graph for this function:



7.7.5.8 template<class... Ts> const std::tuple< Ts... > & prior_hessian::CompositeDist::get_dist_tuple ( ) const

Definition at line 930 of file CompositeDist.h.

Referenced by initialize().

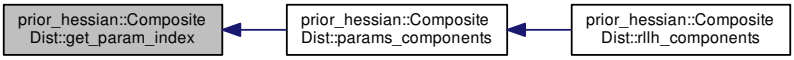
Here is the caller graph for this function:

7.7.5.9 IdxT prior_hessian::CompositeDist::get_param_index (const std::string & *name*) const

Definition at line 221 of file CompositeDist.cpp.

Referenced by params_components().

Here is the caller graph for this function:



7.7.5.10 double prior_hessian::CompositeDist::get_param_value (const std::string & *name*) const

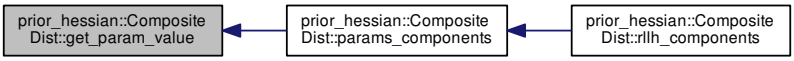
Definition at line 209 of file CompositeDist.cpp.

References params().

Referenced by params_components().

Here is the call graph for this function:

Here is the caller graph for this function:

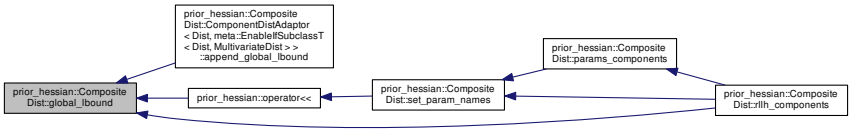


7.7.5.11 VecT prior_hessian::CompositeDist::global_lbound ( ) const [inline]

Definition at line 145 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_global_lbound(), prior_hessian::operator<<(), and rllh_components().

Here is the caller graph for this function:

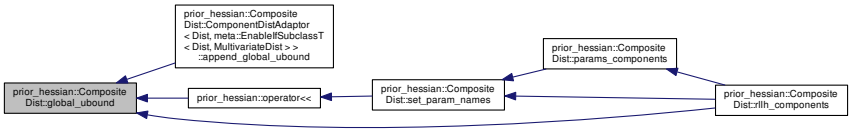


7.7.5.12 VecT prior_hessian::CompositeDist::global_ubound ( ) const [inline]

Definition at line 146 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_global_ubound(), prior_hessian::operator<<(), and rllh_components().

Here is the caller graph for this function:



7.7.5.13 VecT prior_hessian::CompositeDist::grad (const VecT & *u*) const [inline]

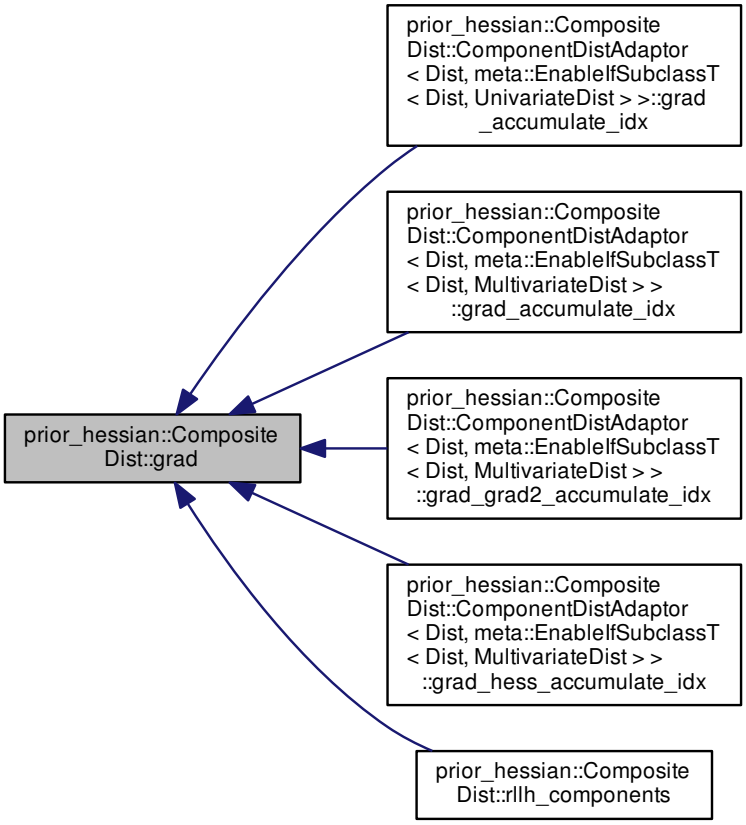
Definition at line 190 of file CompositeDist.h.

References num_dim().

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::grad_accumulate_idx(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad_accumulate_idx(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad_grad2_accumulate_idx(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad_hess_accumulate_idx(), and rllh_components().

Here is the call graph for this function:

Here is the caller graph for this function:



7.7.5.14 VecT prior_hessian::CompositeDist::grad2 (const VecT & *u*) const [inline]

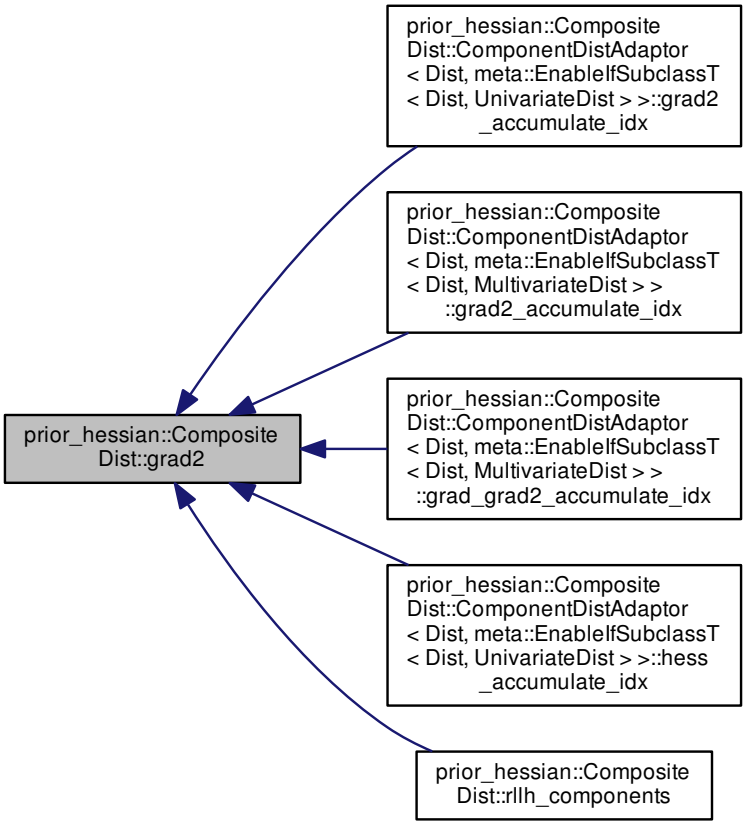
Definition at line 197 of file CompositeDist.h.

References num_dim().

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::grad2_accumulate_idx(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad2_accumulate_idx(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad_grad2_accumulate_idx(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::hess_accumulate_idx(), and rllh_components().

Here is the call graph for this function:

Here is the caller graph for this function:



7.7.5.15 void prior_hessian::CompositeDist::grad2_accumulate (const VecT & *theta,* VecT & *grad2*) const [inline]

Definition at line 219 of file CompositeDist.h.

Referenced by rllh_components().

Here is the caller graph for this function:

7.7.5.16 void prior_hessian::CompositeDist::grad_accumulate (const VecT & *theta,* VecT & *grad*) const [inline]

Definition at line 218 of file CompositeDist.h.

Referenced by rllh_components().

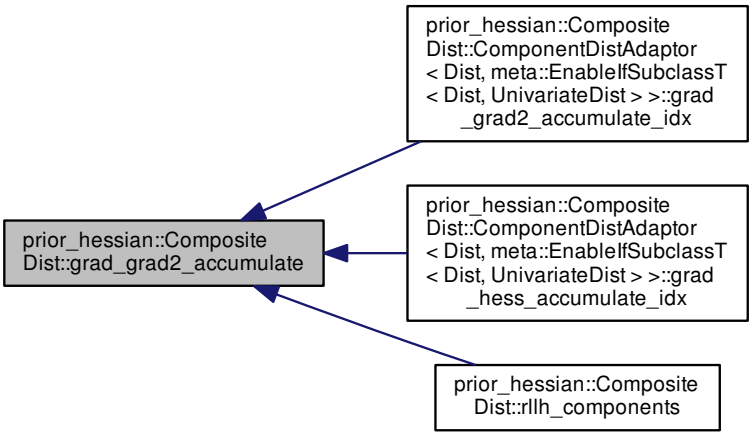
Here is the caller graph for this function:

7.7.5.17 void prior_hessian::CompositeDist::grad_grad2_accumulate (const VecT & *theta,* VecT & *grad,* VecT & *grad2*) const [inline]

Definition at line 221 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::grad_grad2_accumulate_idx(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::grad_hess_accumulate_idx(), and rllh_components().

Here is the caller graph for this function:

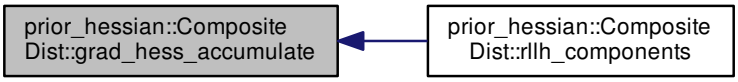


7.7.5.18 void prior_hessian::CompositeDist::grad_hess_accumulate (const VecT & *theta,* VecT & *grad,* MatT & *hess*) const [inline]

Definition at line 224 of file CompositeDist.h.

Referenced by rllh_components().

Here is the caller graph for this function:

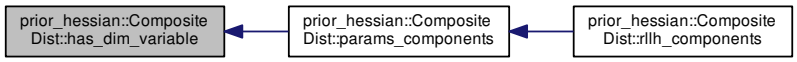


7.7.5.19 bool prior_hessian::CompositeDist::has_dim_variable (const std::string & *name*) const

Definition at line 169 of file CompositeDist.cpp.

Referenced by params_components().

Here is the caller graph for this function:

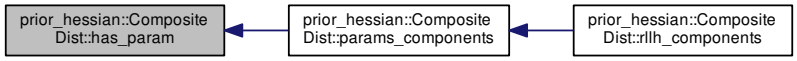


7.7.5.20 bool prior_hessian::CompositeDist::has_param (const std::string & *name*) const

Definition at line 203 of file CompositeDist.cpp.

Referenced by params_components().

Here is the caller graph for this function:



7.7.5.21 MatT prior_hessian::CompositeDist::hess (const VecT & *u*) const [inline]

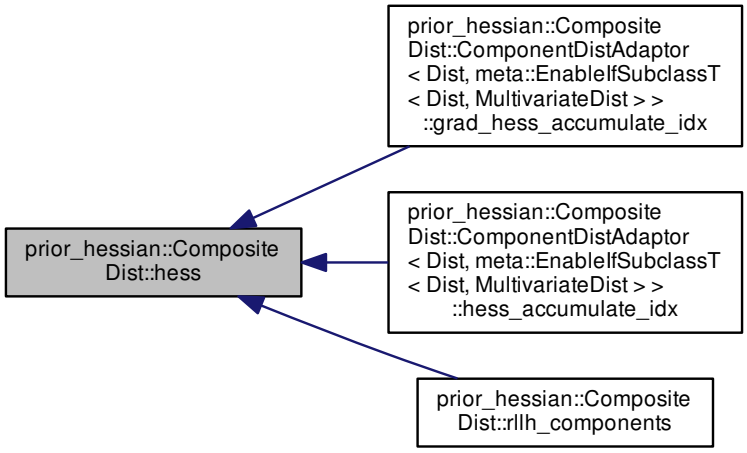
Definition at line 205 of file CompositeDist.h.

References num_dim().

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::grad_hess_accumulate_idx(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::hess_accumulate_idx(), and rllh_components().

Here is the call graph for this function:

Here is the caller graph for this function:



7.7.5.22 void prior_hessian::CompositeDist::hess_accumulate (const VecT & *theta,* MatT & *hess*) const [inline]

Definition at line 220 of file CompositeDist.h.

Referenced by rllh_components().

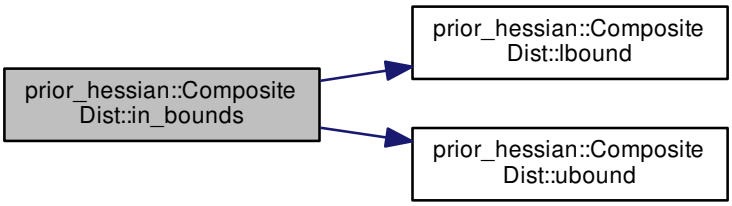
Here is the caller graph for this function:

7.7.5.23 bool prior_hessian::CompositeDist::in_bounds (const VecT & *u*) const [inline]

Definition at line 147 of file CompositeDist.h.

References lbound(), and ubound().

Here is the call graph for this function:

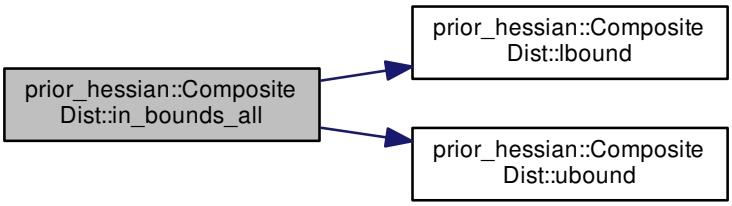


7.7.5.24 bool prior_hessian::CompositeDist::in_bounds_all (const MatT & *u*) const [inline]

Definition at line 149 of file CompositeDist.h.

References lbound(), and ubound().

Here is the call graph for this function:

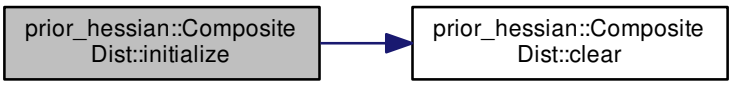


7.7.5.25 void prior_hessian::CompositeDist::initialize ( ) [inline]

Definition at line 89 of file CompositeDist.h.

References clear().

Here is the call graph for this function:



7.7.5.26 void prior_hessian::CompositeDist::initialize (const std::tuple<> &) [inline]

Initialize to the empty state.

Definition at line 90 of file CompositeDist.h.

References clear().

Here is the call graph for this function:

7.7.5.27 void prior_hessian::CompositeDist::initialize (std::tuple<> &&) [inline]

Initialize of an empty lvalue tuple produces the empty state.

Definition at line 91 of file CompositeDist.h.

References clear().

Here is the call graph for this function:

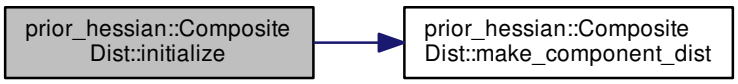
7.7.5.28 template<class... Ts, typename = meta::EnableIfAllAreNotTupleT<Ts...>> void prior_hessian::CompositeDist::initialize (Ts &&... *dists*) [inline]

Initialize of an empty rvalue tuple produces the empty state.

Definition at line 93 of file CompositeDist.h.

References make_component_dist().

Here is the call graph for this function:

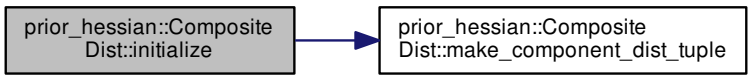


7.7.5.29 template<class... Ts, typename = meta::EnableIfAllAreNotTupleT<Ts...>> void prior_hessian::CompositeDist::initialize (std::tuple< Ts... > && *dist_tuple*) [inline]

Definition at line 100 of file CompositeDist.h.

References make_component_dist_tuple().

Here is the call graph for this function:

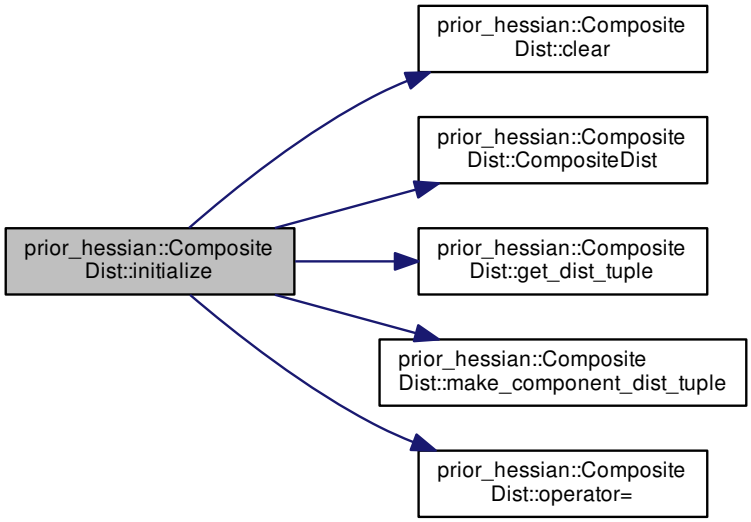


7.7.5.30 template<class... Ts, typename = meta::EnableIfNonEmpty<Ts...>> void prior_hessian::CompositeDist::initialize (const std::tuple< Ts... > & *dist_tuple*) [inline]

Definition at line 107 of file CompositeDist.h.

References clear(), CompositeDist(), get_dist_tuple(), make_component_dist_tuple(), and operator=().

Here is the call graph for this function:

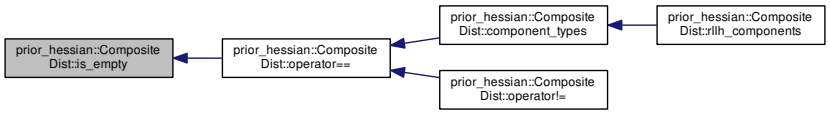


7.7.5.31 bool prior_hessian::CompositeDist::is_empty ( ) const [inline]

Definition at line 124 of file CompositeDist.h.

Referenced by operator==().

Here is the caller graph for this function:

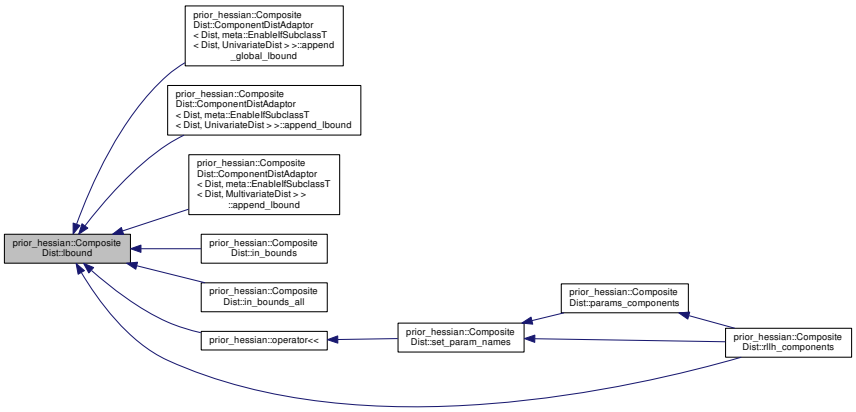


7.7.5.32 VecT prior_hessian::CompositeDist::lbound ( ) const [inline]

Definition at line 143 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_global_lbound(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_lbound(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_lbound(), in_bounds(), in_bounds_all(), prior_hessian::operator<<(), and rllh_components().

Here is the caller graph for this function:

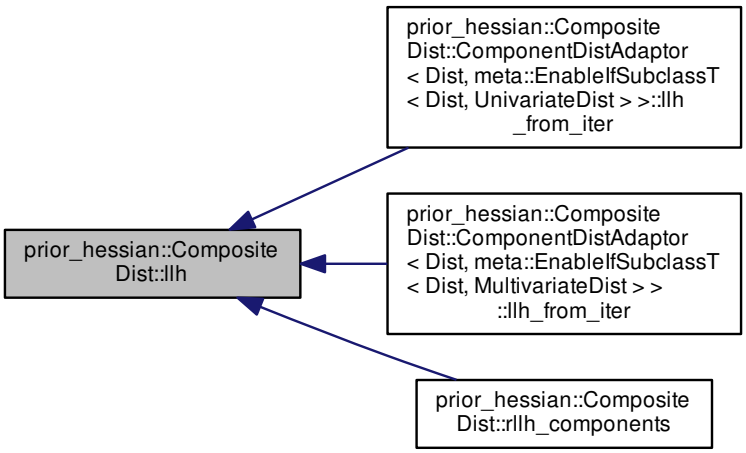


7.7.5.33 double prior_hessian::CompositeDist::llh (const VecT & *u*) const [inline]

Definition at line 188 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::llh_from_iter(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIf↩SubclassT< Dist, MultivariateDist > >::llh_from_iter(), and rllh_components().

Here is the caller graph for this function:



7.7.5.34 VecT prior_hessian::CompositeDist::llh_components (const VecT & *u*) const [inline]

Definition at line 249 of file CompositeDist.h.

Referenced by rllh_components().

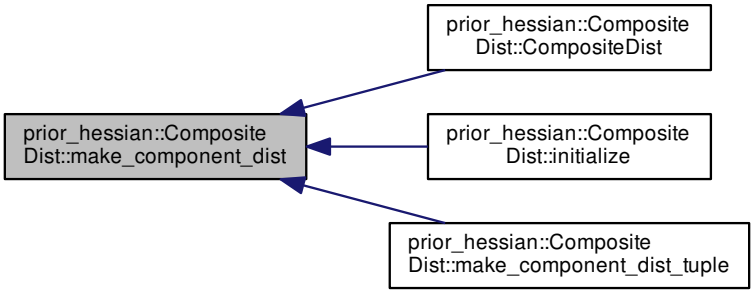
Here is the caller graph for this function:

7.7.5.35 template<class DistT > static meta::ReturnIfInstantiatedFromT<DistT,DistT,ComponentDistAdaptor> prior_hessian::CompositeDist::make_component_dist (DistT && *dist*) [inline], [static]

Definition at line 851 of file CompositeDist.h.

Referenced by CompositeDist(), initialize(), and make_component_dist_tuple().

Here is the caller graph for this function:

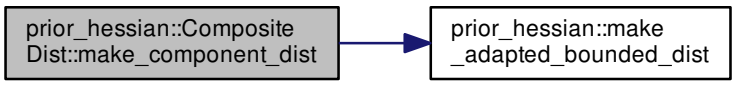


7.7.5.36 template<class DistT > static meta::ReturnIfNotInstantiatedFromT<Component↩ DistT<DistT>,DistT,ComponentDistAdaptor> prior_hessian::CompositeDist::make_component_dist (DistT && *dist*) [inline],[static]

Definition at line 857 of file CompositeDist.h.

References prior_hessian::make_adapted_bounded_dist().

Here is the call graph for this function:

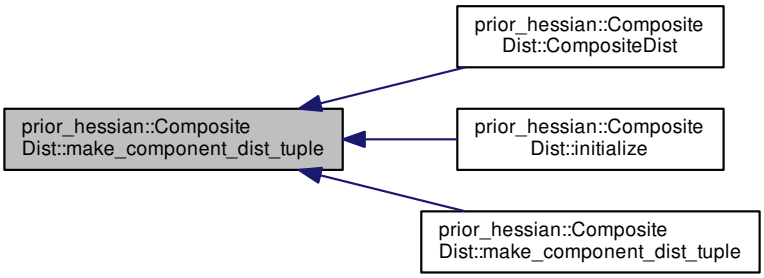


7.7.5.37 template<class... Ts> static std::tuple<ComponentDistT<Ts>...> prior_hessian::↩ CompositeDist::make_component_dist_tuple (const std::tuple< Ts... > & *dists*) [inline], [static]

Definition at line 867 of file CompositeDist.h.

Referenced by CompositeDist(), initialize(), and make_component_dist_tuple().

Here is the caller graph for this function:

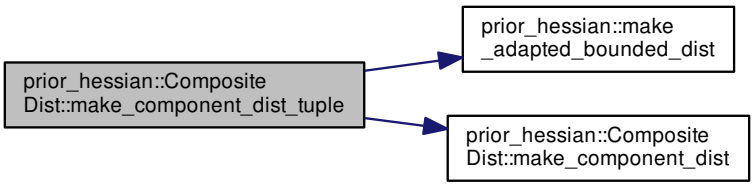


7.7.5.38 template<class... Ts, std::size_t... I> static std::tuple<ComponentDistT<Ts>...> prior_hessian::CompositeDist::make_component_dist_tuple (const std::tuple< Ts... > & *dists*, std::index_sequence< I... >) [inline], [static]

Definition at line 874 of file CompositeDist.h.

References prior_hessian::make_adapted_bounded_dist(), and make_component_dist().

Here is the call graph for this function:



7.7.5.39 template<class... Ts> static std::tuple<ComponentDistT<Ts>...> prior_hessian::CompositeDist::make_component_dist_tuple (std::tuple< Ts... > && *dists*) [inline], [static]

Definition at line 880 of file CompositeDist.h.

References make_component_dist_tuple().

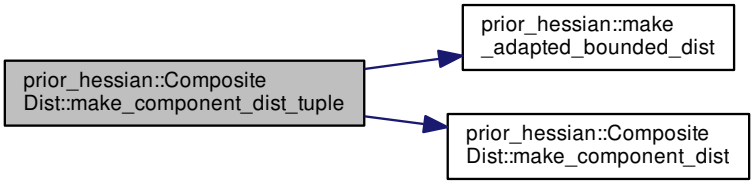
Here is the call graph for this function:

7.7.5.40 template<class... Ts, std::size_t... I> static std::tuple<ComponentDistT<Ts>...> prior_hessian::CompositeDist::make_component_dist_tuple (std::tuple< Ts... > && *dists*, std::index_sequence< I... >) [inline], [static]

Definition at line 887 of file CompositeDist.h.

References prior_hessian::make_adapted_bounded_dist(), and make_component_dist().

Here is the call graph for this function:



7.7.5.41 VecT prior_hessian::CompositeDist::make_zero_grad ( ) const [inline]

Definition at line 226 of file CompositeDist.h.

References num_dim().

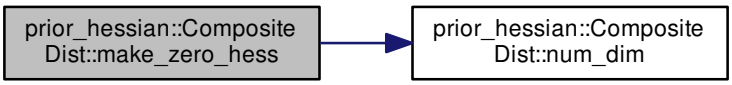
Here is the call graph for this function:

7.7.5.42 MatT prior_hessian::CompositeDist::make_zero_hess ( ) const [inline]

Definition at line 227 of file CompositeDist.h.

References num_dim().

Here is the call graph for this function:

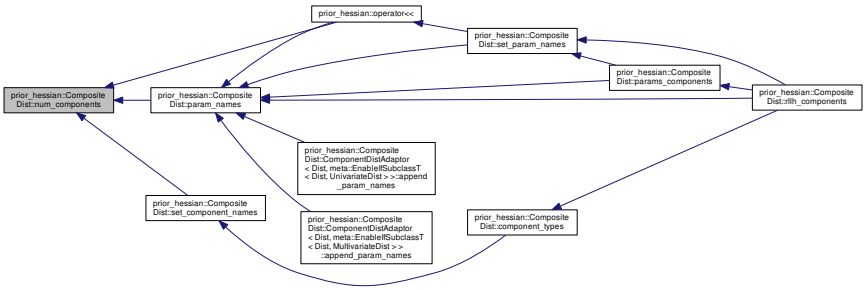


7.7.5.43 IdxT prior_hessian::CompositeDist::num_components ( ) const [inline]

Definition at line 126 of file CompositeDist.h.

Referenced by prior_hessian::operator<<(), param_names(), and set_component_names().

Here is the caller graph for this function:

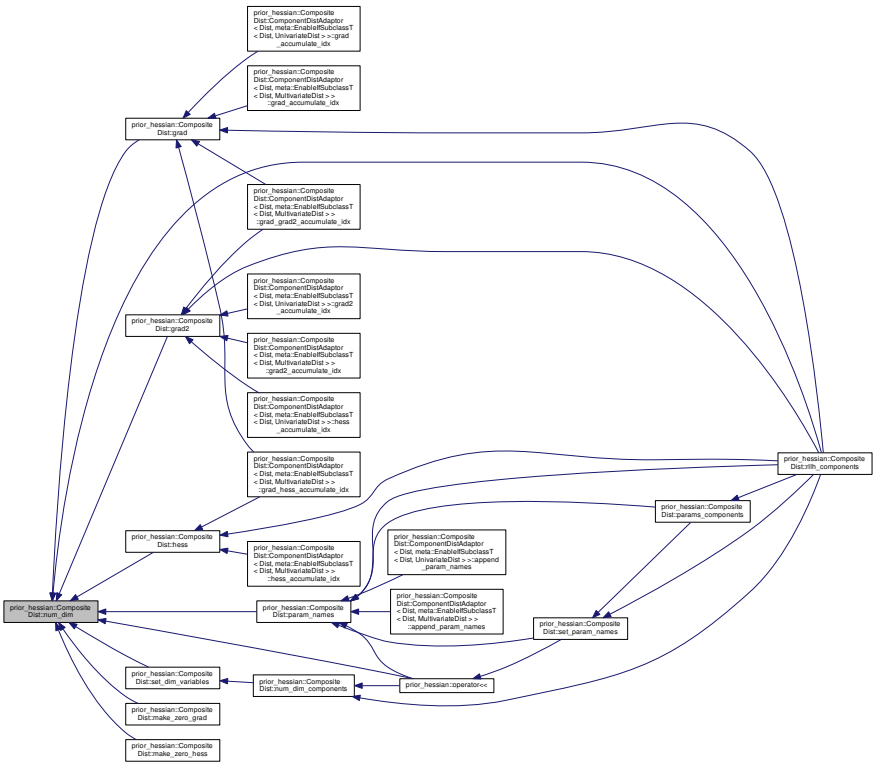


7.7.5.44 IdxT prior_hessian::CompositeDist::num_dim ( ) const [inline]

Definition at line 136 of file CompositeDist.h.

Referenced by grad(), grad2(), hess(), make_zero_grad(), make_zero_hess(), prior_hessian::operator<<(), param_↩ names(), rllh_components(), and set_dim_variables().

Here is the caller graph for this function:



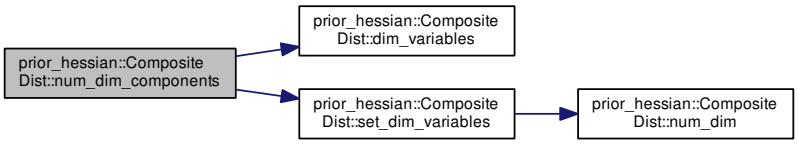
7.7.5.45 UVecT prior_hessian::CompositeDist::num_dim_components ( ) const [inline]

Definition at line 137 of file CompositeDist.h.

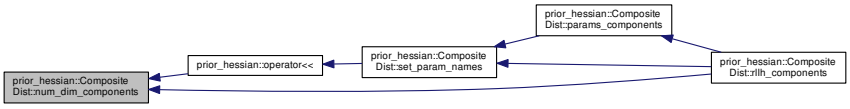
References dim_variables(), and set_dim_variables().

Referenced by prior_hessian::operator<<(), and rllh_components().

Here is the call graph for this function:



Here is the caller graph for this function:

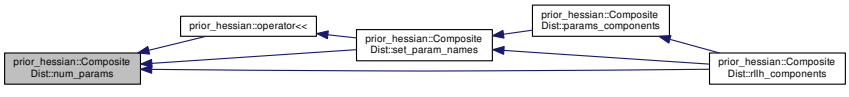


7.7.5.46 IdxT prior_hessian::CompositeDist::num_params ( ) const [inline]

Definition at line 159 of file CompositeDist.h.

Referenced by prior_hessian::operator<<(), rllh_components(), and set_param_names().

Here is the caller graph for this function:

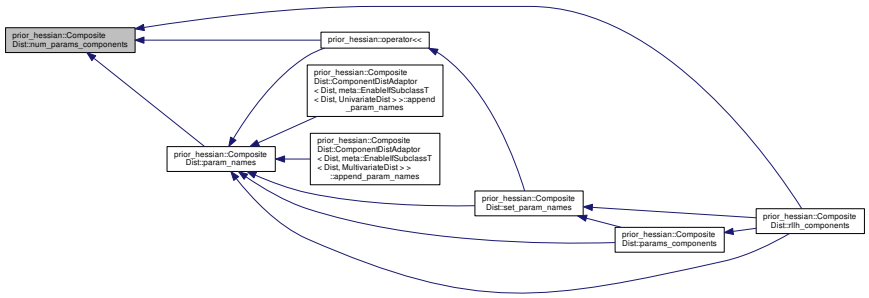


7.7.5.47 UVecT prior_hessian::CompositeDist::num_params_components ( ) const [inline]

Definition at line 160 of file CompositeDist.h.

Referenced by prior_hessian::operator<<(), param_names(), and rllh_components().

Here is the caller graph for this function:



7.7.5.48 prior_hessian::CompositeDist::operator bool ( ) const [inline]

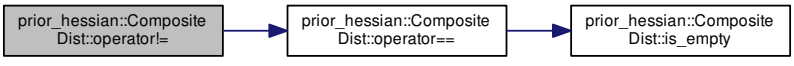
Definition at line 125 of file CompositeDist.h.

7.7.5.49 bool prior_hessian::CompositeDist::operator!= (const CompositeDist & *o*) const [inline]

Definition at line 133 of file CompositeDist.h.

References operator==().

Here is the call graph for this function:



7.7.5.50 CompositeDist & prior_hessian::CompositeDist::operator= (const CompositeDist & *o*)

Definition at line 44 of file CompositeDist.cpp.

Referenced by initialize().

Here is the caller graph for this function:

7.7.5.51 CompositeDist & prior_hessian::CompositeDist::operator= (CompositeDist && *o*)

Definition at line 57 of file CompositeDist.cpp.

7.7.5.52 bool prior_hessian::CompositeDist::operator== (const CompositeDist & *o*) const

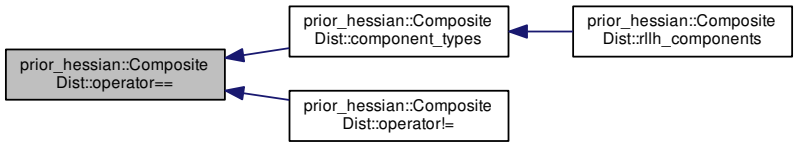
Definition at line 70 of file CompositeDist.cpp.

References is_empty().

Referenced by component_types(), and operator!=().

Here is the call graph for this function:

Here is the caller graph for this function:



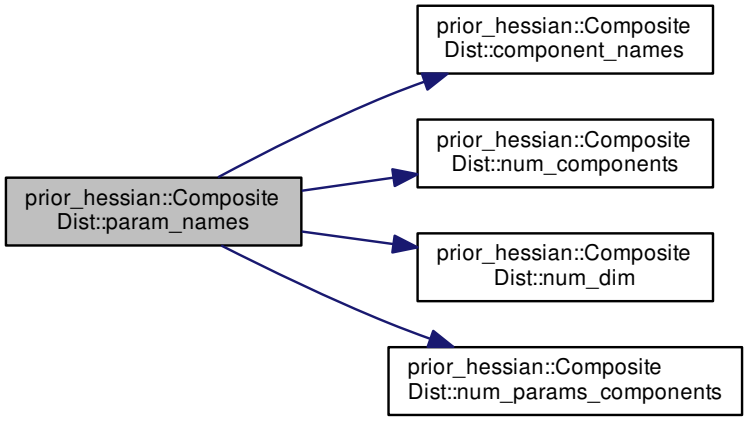
7.7.5.53 const StringVecT & prior_hessian::CompositeDist::param_names ( ) const

Definition at line 120 of file CompositeDist.cpp.

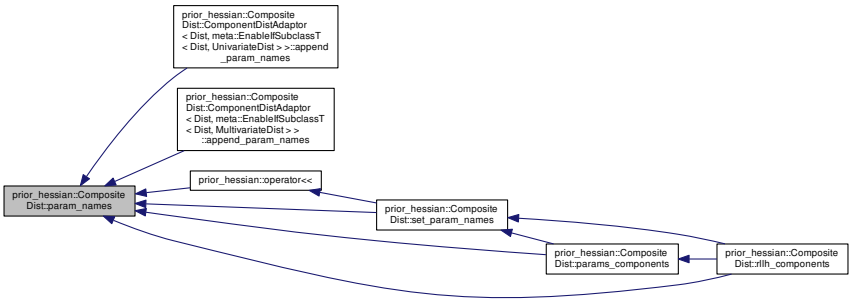
References component_names(), num_components(), num_dim(), and num_params_components().

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_param_names(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_param_names(), prior_hessian::operator<<(), params_components(), rllh_components(), and set_param_names().

Here is the call graph for this function:



Here is the caller graph for this function:

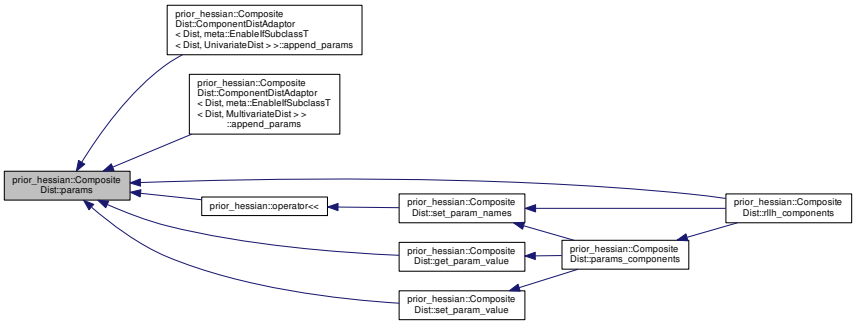


7.7.5.54 VecT prior_hessian::CompositeDist::params ( ) const [inline]

Definition at line 161 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_params(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_params(), get_param_value(), prior_hessian::operator<<(), rllh_components(), and set_param_value().

Here is the caller graph for this function:



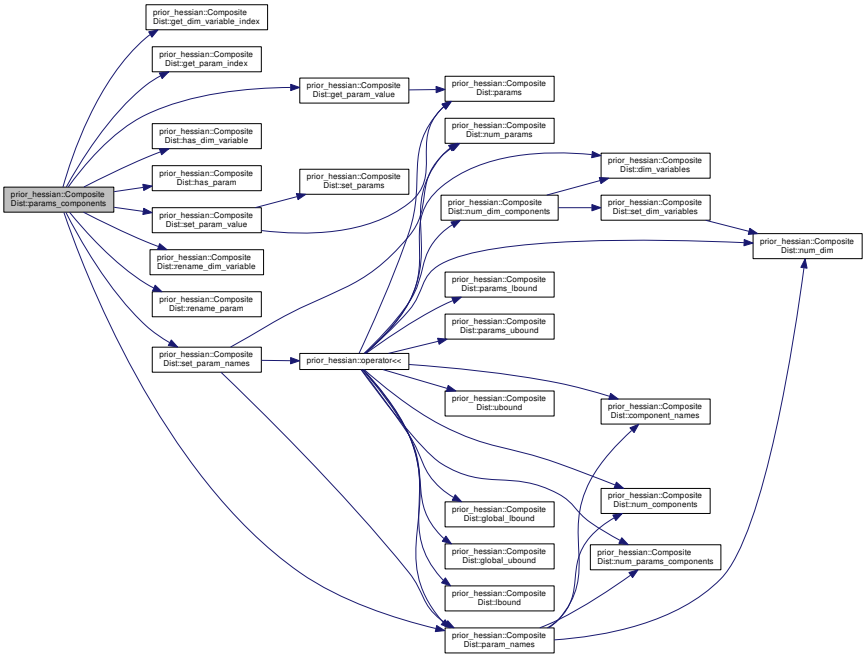
7.7.5.55 std::vector<VecT> prior_hessian::CompositeDist::params_components ( ) const [inline]

Definition at line 166 of file CompositeDist.h.

References get_dim_variable_index(), get_param_index(), get_param_value(), has_dim_variable(), has_param(), param_names(), rename_dim_variable(), rename_param(), set_param_names(), and set_param_value().

Referenced by rllh_components().

Here is the call graph for this function:



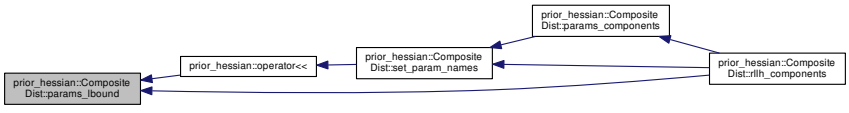
Here is the caller graph for this function:

7.7.5.56 VecT prior_hessian::CompositeDist::params_lbound ( ) const [inline]

Definition at line 164 of file CompositeDist.h.

Referenced by prior_hessian::operator<<(), and rllh_components().

Here is the caller graph for this function:

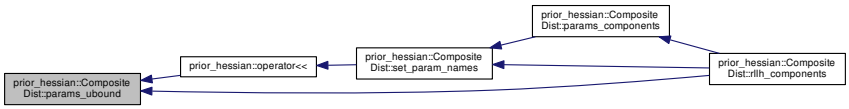


7.7.5.57 VecT prior_hessian::CompositeDist::params_ubound ( ) const [inline]

Definition at line 165 of file CompositeDist.h.

Referenced by prior_hessian::operator<<(), and rllh_components().

Here is the caller graph for this function:

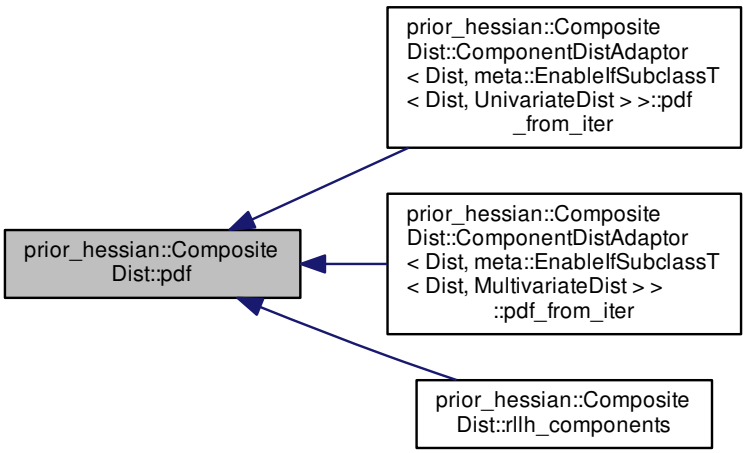


7.7.5.58 double prior_hessian::CompositeDist::pdf (const VecT & *u*) const [inline]

Definition at line 187 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::pdf_from_iter(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIf↩SubclassT< Dist, MultivariateDist > >::pdf_from_iter(), and rllh_components().

Here is the caller graph for this function:

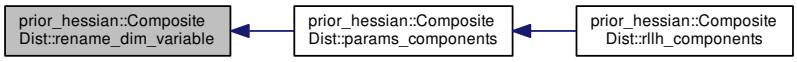


7.7.5.59 void prior_hessian::CompositeDist::rename_dim_variable (const std::string & *old_name,* std::string *new_name*)

Definition at line 187 of file CompositeDist.cpp.

Referenced by params_components().

Here is the caller graph for this function:

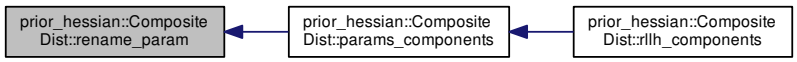


7.7.5.60 void prior_hessian::CompositeDist::rename_param (const std::string & *old_name,* std::string *new_name*)

Definition at line 247 of file CompositeDist.cpp.

Referenced by params_components().

Here is the caller graph for this function:

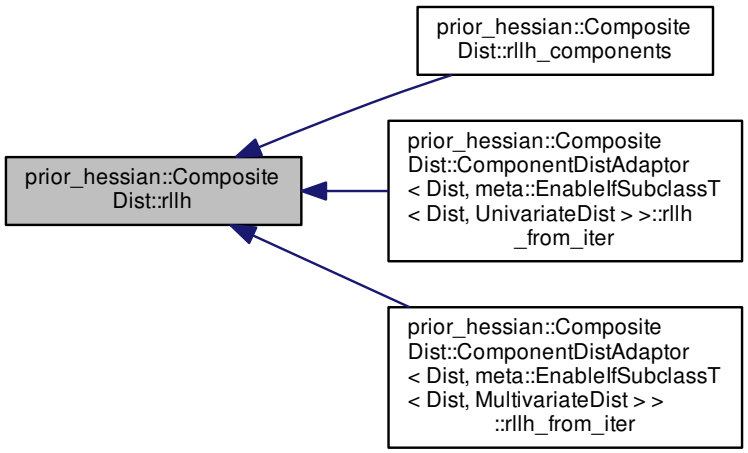


7.7.5.61 double prior_hessian::CompositeDist::rllh (const VecT & *u*) const [inline]

Definition at line 189 of file CompositeDist.h.

Referenced by rllh_components(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIf↩ SubclassT< Dist, UnivariateDist > >::rllh_from_iter(), and prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::rllh_from_iter().

Here is the caller graph for this function:

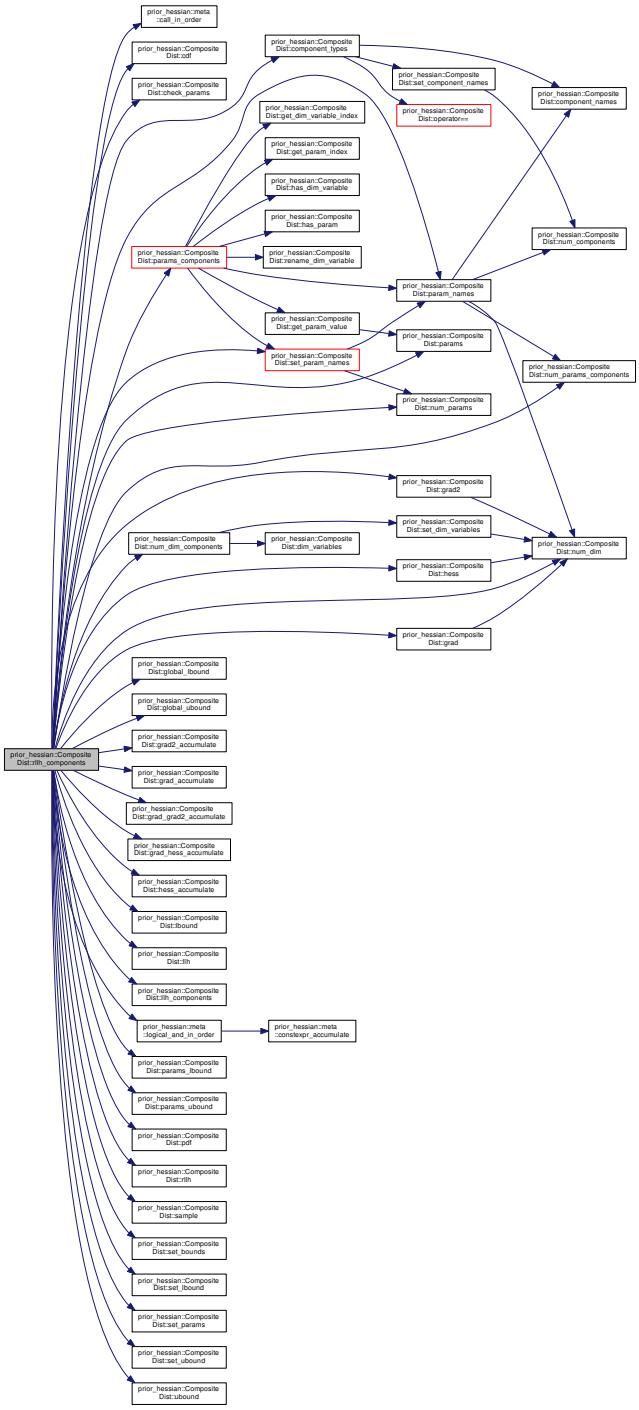


7.7.5.62 VecT prior_hessian::CompositeDist::rllh_components (const VecT & *u*) const [inline]

Definition at line 250 of file CompositeDist.h.

References prior_hessian::meta::call_in_order(), cdf(), check_params(), component_types(), global_lbound(), global_ubound(), grad(), grad2(), grad2_accumulate(), grad_accumulate(), grad_grad2_accumulate(), grad_hess_accumulate(), hess(), hess_accumulate(), lbound(), llh(), llh_components(), prior_hessian::meta::logical_and_in_order(), num_dim(), num_dim_components(), num_params(), num_params_components(), param_names(), params(), params_components(), params_lbound(), params_ubound(), pdf(), rllh(), sample(), set_bounds(), set_lbound(), set_param_names(), set_params(), set_ubound(), and ubound().

Here is the call graph for this function:

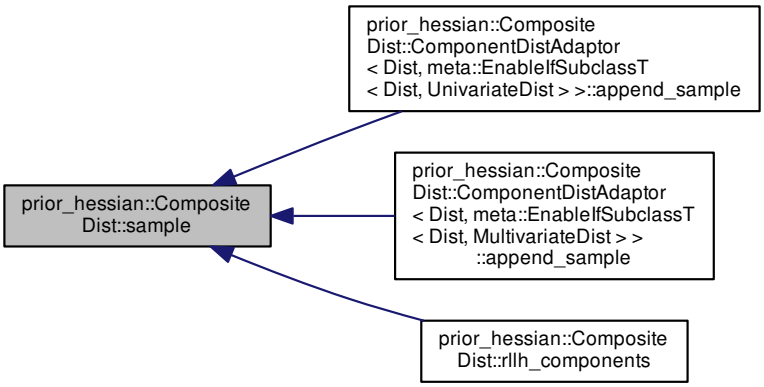


7.7.5.63 VecT prior_hessian::CompositeDist::sample (AnyRngT & *rng*) const [inline]

Definition at line 229 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_sample(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIf↩SubclassT< Dist, MultivariateDist > >::append_sample(), and rllh_components().

Here is the caller graph for this function:



7.7.5.64 VecT prior_hessian::CompositeDist::sample (AnyRngT && *rng*) const [inline]

Definition at line 230 of file CompositeDist.h.

7.7.5.65 MatT prior_hessian::CompositeDist::sample (AnyRngT & *rng*, IdxT *num_samples*) const [inline]

Definition at line 231 of file CompositeDist.h.

7.7.5.66 MatT prior_hessian::CompositeDist::sample (AnyRngT && *rng*, IdxT *num_samples*) const [inline]

Definition at line 232 of file CompositeDist.h.

7.7.5.67 template<class RngT > VecT prior_hessian::CompositeDist::sample (RngT && *rng*) const [inline]

Definition at line 235 of file CompositeDist.h.

7.7.5.68 template<class RngT > MatT prior_hessian::CompositeDist::sample (RngT && *rng*, IdxT *num_samples*) const [inline]

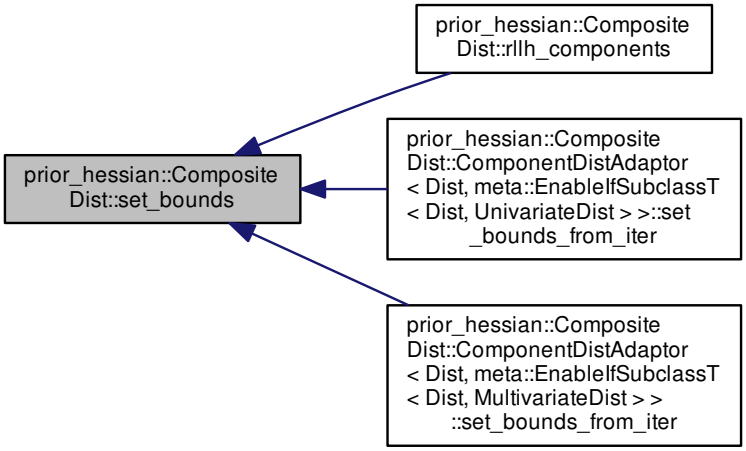
Definition at line 242 of file CompositeDist.h.

7.7.5.69 void prior_hessian::CompositeDist::set_bounds (const VecT & *new_lbound,* const VecT & *new_ubound*) [inline]

Definition at line 156 of file CompositeDist.h.

Referenced by rllh_components(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIf↩ SubclassT< Dist, UnivariateDist > >::set_bounds_from_iter(), and prior_hessian::CompositeDist::ComponentDist↩ Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::set_bounds_from_iter().

Here is the caller graph for this function:



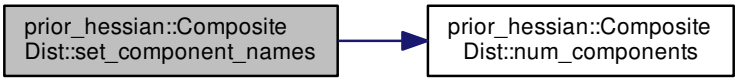
7.7.5.70 template<class StringVec > void prior_hessian::CompositeDist::set_component_names (StringVec && *names*)

Definition at line 943 of file CompositeDist.h.

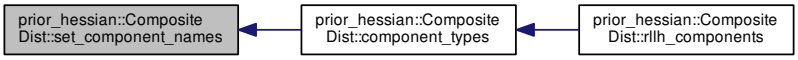
References num_components().

Referenced by component_types().

Here is the call graph for this function:



Here is the caller graph for this function:



7.7.5.71 template<class StringVec > void prior_hessian::CompositeDist::set_dim_variables (StringVec && *vars*)

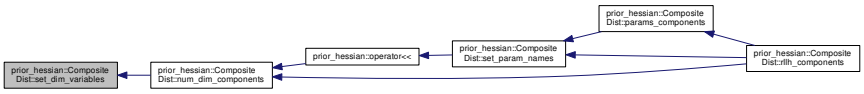
Definition at line 955 of file CompositeDist.h.

References num_dim().

Referenced by num_dim_components().

Here is the call graph for this function:

Here is the caller graph for this function:

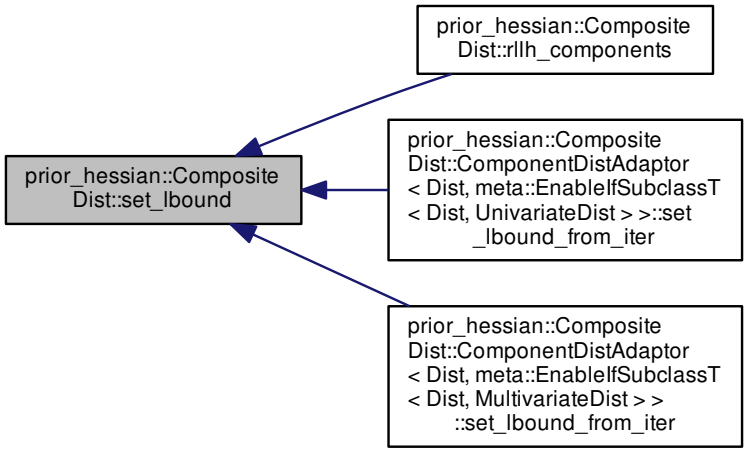


7.7.5.72 void prior_hessian::CompositeDist::set_lbound (const VecT & *new_bound*) [inline]

Definition at line 154 of file CompositeDist.h.

Referenced by rllh_components(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIf↩ SubclassT< Dist, UnivariateDist > >::set_lbound_from_iter(), and prior_hessian::CompositeDist::ComponentDist↩ Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::set_lbound_from_iter().

Here is the caller graph for this function:



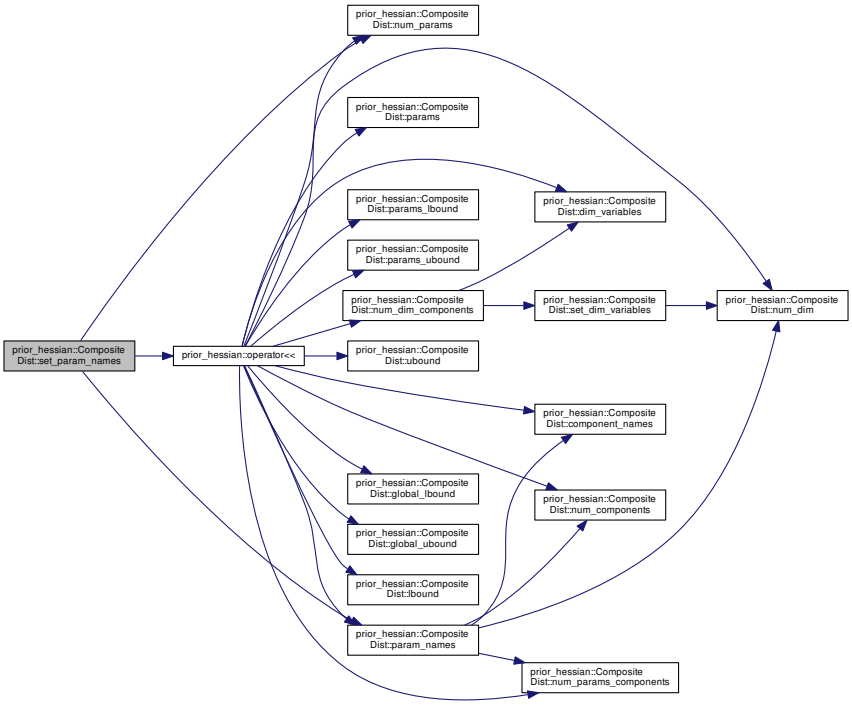
7.7.5.73 template<class StringVec > void prior_hessian::CompositeDist::set_param_names (StringVec && *vars*)

Definition at line 967 of file CompositeDist.h.

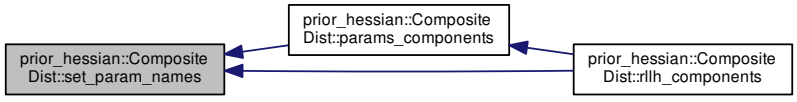
References num_params(), prior_hessian::operator<<(), and param_names().

Referenced by params_components(), and rllh_components().

Here is the call graph for this function:



Here is the caller graph for this function:



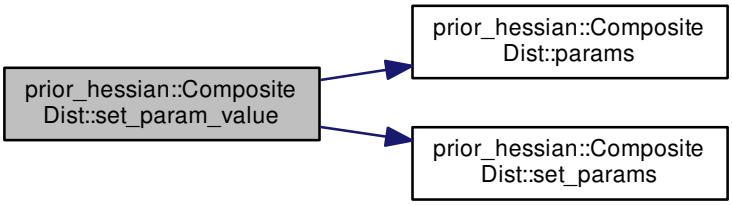
7.7.5.74 void prior_hessian::CompositeDist::set_param_value (const std::string & *name*, double *value*)

Definition at line 233 of file CompositeDist.cpp.

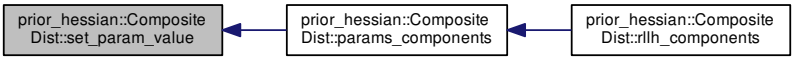
References params(), and set_params().

Referenced by params_components().

Here is the call graph for this function:



Here is the caller graph for this function:

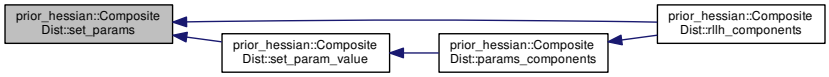


7.7.5.75 void prior_hessian::CompositeDist::set_params (const VecT & *new_params*) [inline]

Definition at line 162 of file CompositeDist.h.

Referenced by rllh_components(), and set_param_value().

Here is the caller graph for this function:

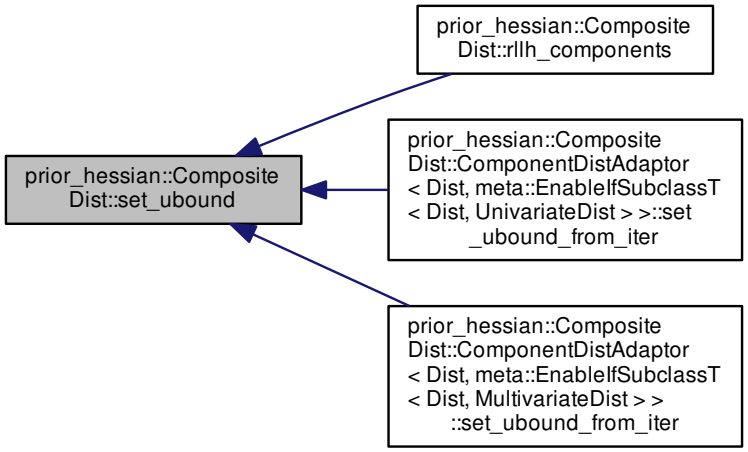


7.7.5.76 void prior_hessian::CompositeDist::set_ubound (const VecT & *new_bound*) [inline]

Definition at line 155 of file CompositeDist.h.

Referenced by rllh_components(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIf↩ SubclassT< Dist, UnivariateDist > >::set_ubound_from_iter(), and prior_hessian::CompositeDist::ComponentDist↩ Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::set_ubound_from_iter().

Here is the caller graph for this function:

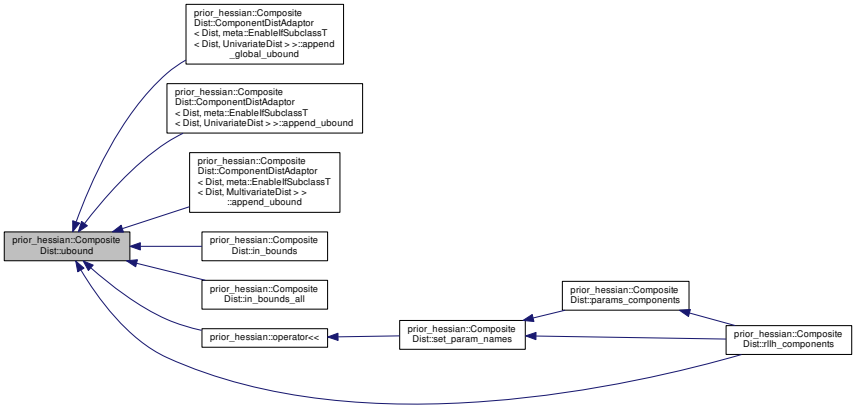


7.7.5.77 VecT prior_hessian::CompositeDist::ubound ( ) const [inline]

Definition at line 144 of file CompositeDist.h.

Referenced by prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_global_ubound(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >::append_ubound(), prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >::append_ubound(), in_bounds(), in_bounds_all(), prior_hessian::operator<<(), and rllh_components().

Here is the caller graph for this function:



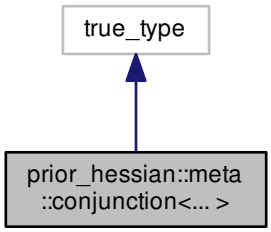
The documentation for this class was generated from the following files:

- CompositeDist.h
- CompositeDist.cpp

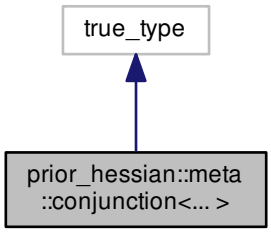
7.8 prior_hessian::meta::conjunction<... > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/Meta.h>`

Inheritance diagram for prior_hessian::meta::conjunction<... >:



Collaboration diagram for prior_hessian::meta::conjunction<... >:



7.8.1 Detailed Description

template<class...>
struct prior_hessian::meta::conjunction<... >

Definition at line 66 of file Meta.h.

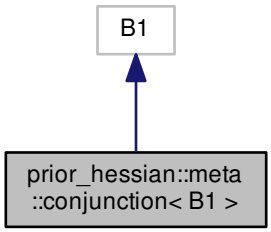
The documentation for this struct was generated from the following file:

- Meta.h

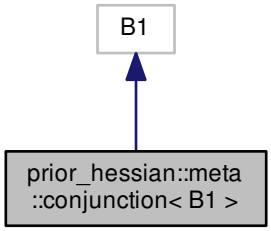
7.9 prior_hessian::meta::conjunction< B1 > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/Meta.h>`

Inheritance diagram for prior_hessian::meta::conjunction< B1 >:



Collaboration diagram for prior_hessian::meta::conjunction< B1 >:



7.9.1 Detailed Description

template<class B1>
struct prior_hessian::meta::conjunction< B1 >

Definition at line 67 of file Meta.h.

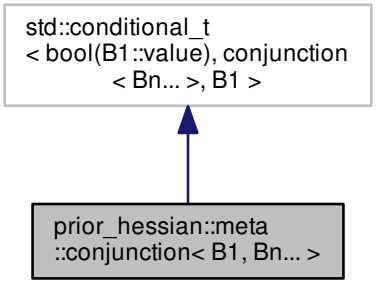
The documentation for this struct was generated from the following file:

- Meta.h

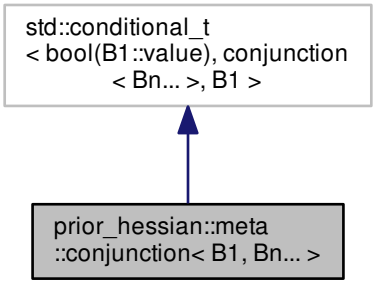
7.10 prior_hessian::meta::conjunction< B1, Bn... > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/Meta.h>`

Inheritance diagram for prior_hessian::meta::conjunction< B1, Bn... >:



Collaboration diagram for prior_hessian::meta::conjunction< B1, Bn... >:



7.10.1 Detailed Description

template<class B1, class... Bn>
struct prior_hessian::meta::conjunction< B1, Bn... >

Definition at line 69 of file Meta.h.

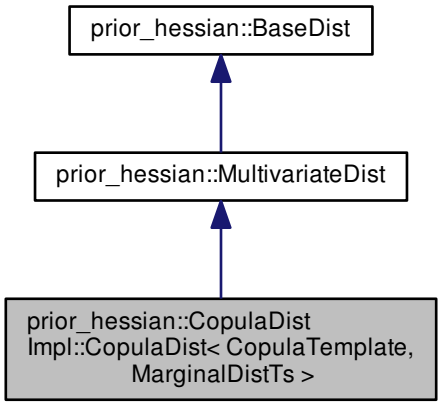
The documentation for this struct was generated from the following file:

- Meta.h

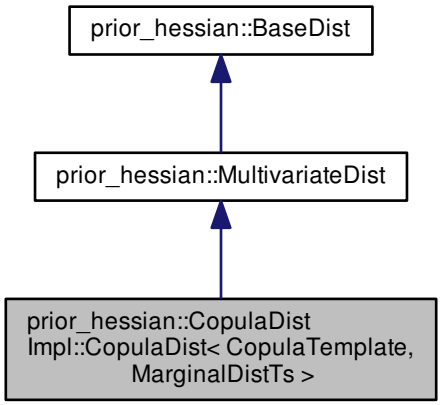
7.11 prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ CopulaDist.h>`

Inheritance diagram for prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >:



Collaboration diagram for prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >:



Public Types

- using NdimVecT = arma::Col< double >::fixed< _num_dim >
- using NdimMatT = arma::Mat< double >::fixed< _num_dim, _num_dim >
- using NparamsVecT = arma::Col< double >
- using MarginalDistTupleT = std::tuple< MarginalDistTs... >
- using CopulaT = CopulaTemplate< _num_dim >
- template<size_t I>
  using MarginalDistT = std::tuple_element< I, MarginalDistTupleT >

Public Member Functions

- CopulaDist ()
- template<class Copula , class... DistTs, std::enable_if_t< sizeof...(DistTs)==sizeof...(MarginalDistTs), bool > Enable = true>
  CopulaDist (Copula &&copula, DistTs &&...dists)
- void initialize_copula (const CopulaT &_copula)
- void initialize_marginals (const MarginalDistTupleT &dists)
- NdimVecT lbound () const
- NdimVecT ubound () const
- template<class Vec , class Vec2 >
  void set_bounds (const Vec &lbound, const Vec2 &ubound)
- template<class Vec >
  void set_lbound (const Vec &lbound)
- template<class Vec >
  void set_ubound (const Vec &ubound)
- bool operator== (const CopulaDist< CopulaTemplate, MarginalDistTs... > &o) const
- bool operator!= (const CopulaDist< CopulaTemplate, MarginalDistTs... > &o) const
- NparamsVecT params () const
- double get_copula_theta () const
- void set_copula_theta (double theta)
- template<class Vec >
  void set_params (const Vec &params)
- template<class Vec >
  double cdf (const Vec &x) const
- template<class Vec >
  double pdf (const Vec &x) const
- template<class Vec >
  double llh (const Vec &x) const
- template<class Vec >
  double rllh (const Vec &x) const
- template<class Vec >
  NdimVecT grad (const Vec &x) const
- template<class Vec >
  NdimVecT grad2 (const Vec &x) const
- template<class Vec >
  NdimMatT hess (const Vec &x) const
- template<class Vec , class Vec2 >
  void grad_grad2_accumulate (const Vec &x, Vec2 &g, Vec2 &g2) const
- template<class Vec , class Vec2 , class Mat >
  void grad_hess_accumulate (const Vec &x, Vec2 &g, Mat &hess) const
- template<class RngT >
  NdimVecT sample (RngT &rng) const
- template<class IterT >
  void set_params_iter (IterT &params)

Static Public Member Functions

- static IdxT num_params ()
- static constexpr IdxT num_components ()
- static constexpr IdxT num_dim ()
- template<class Vec >
  static bool check_params (const Vec &params)
- static bool check_copula_theta (double theta)
- static const StringVecT & param_names ()
- static const NparamsVecT & param_lbound ()
- static const NparamsVecT & param_ubound ()
- static const NdimVecT & global_lbound ()
- static const NdimVecT & global_ubound ()
- template<class IterT >
  static bool check_params_iter (IterT &params)

Static Protected Member Functions

- template<class Vec >
  static void check_bounds (const Vec &lbound, const Vec &ubound)

7.11.1 Detailed Description

template<template< int > class CopulaTemplate, class... MarginalDistTs>
class prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >

Definition at line 22 of file CopulaDist.h.

7.11.2 Member Typedef Documentation

7.11.2.1 template<template< int > class CopulaTemplate, class... MarginalDistTs> using prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::CopulaT = CopulaTemplate<_num_dim>

Definition at line 41 of file CopulaDist.h.

7.11.2.2 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<size_t I> using prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::MarginalDistT = std::tuple_element<I,MarginalDistTupleT>

Definition at line 44 of file CopulaDist.h.

7.11.2.3 template<template< int > class CopulaTemplate, class... MarginalDistTs> using prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::MarginalDistTupleT = std::tuple<MarginalDistTs...>

Definition at line 40 of file CopulaDist.h.

7.11.2.4 template<template< int > class CopulaTemplate, class... MarginalDistTs> using prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::NdimMatT = arma::Mat<double>::fixed<_num_dim,_num_dim>

Definition at line 32 of file CopulaDist.h.

7.11.2.5 template<template< int > class CopulaTemplate, class... MarginalDistTs> using prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::NdimVecT = arma::Col<double>::fixed<_num_dim>

Definition at line 31 of file CopulaDist.h.

7.11.2.6 template<template< int > class CopulaTemplate, class... MarginalDistTs> using prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::NparamsVecT = arma::Col<double>

Definition at line 37 of file CopulaDist.h.

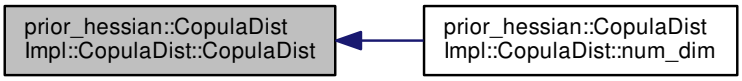
7.11.3 Constructor & Destructor Documentation

7.11.3.1 template<template< int > class CopulaTemplate, class... MarginalDistTs> prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::CopulaDist ( )

Definition at line 242 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim().

Here is the caller graph for this function:



7.11.3.2 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Copula , class... DistTs, std::enable_if_t< sizeof...(DistTs)==sizeof...(MarginalDistTs), bool > Enable> prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::CopulaDist (Copula && *copula,* DistTs &&... *dists*)

Definition at line 233 of file CopulaDist.h.

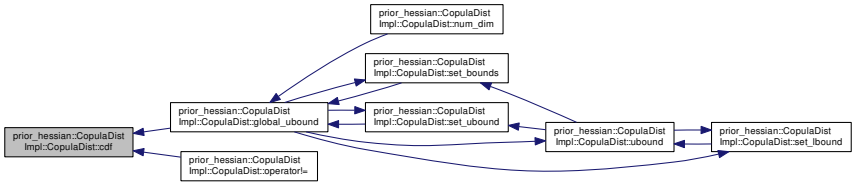
7.11.4 Member Function Documentation

7.11.4.1 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > double prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::cdf (const Vec & *x*) const

Definition at line 385 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:



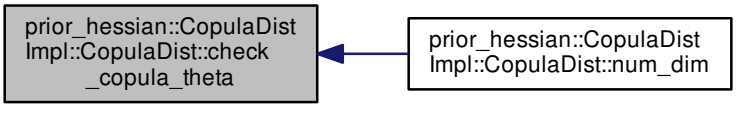
7.11.4.2 template<class Vec > static void prior_hessian::MultivariateDist::check_bounds (const Vec & *lbound,* const Vec & *ubound*) [inline],[static],[protected],[inherited]

Definition at line 57 of file MultivariateDist.h.

7.11.4.3 template<template< int > class CopulaTemplate, class... MarginalDistTs> static bool prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::check_copula_theta (double *theta*) [static]

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim().

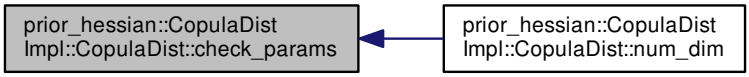
Here is the caller graph for this function:



7.11.4.4 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > static bool prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::check_params (const Vec & *params*) [static]

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim().

Here is the caller graph for this function:

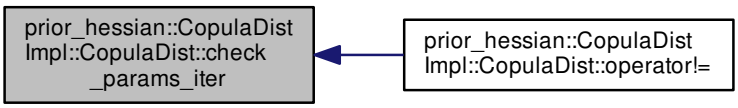


7.11.4.5 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class IterT > bool prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::check_params_iter (IterT & *params*) [static]

Definition at line 368 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

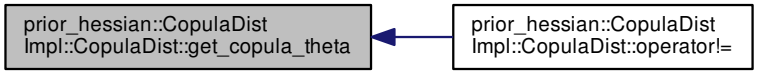


7.11.4.6 template<template< int > class CopulaTemplate, class... MarginalDistTs> double prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::get_copula_theta ( ) const

Definition at line 346 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

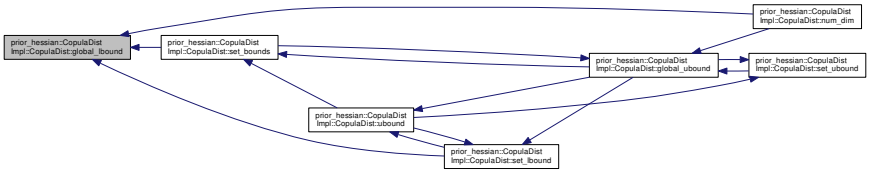


7.11.4.7 template<template< int > class CopulaTemplate, class... MarginalDistTs> const CopulaDist< CopulaTemplate, MarginalDistTs... >::NdimVecT & prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_lbound ( ) [static]

Definition at line 536 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_bounds(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_lbound().

Here is the caller graph for this function:



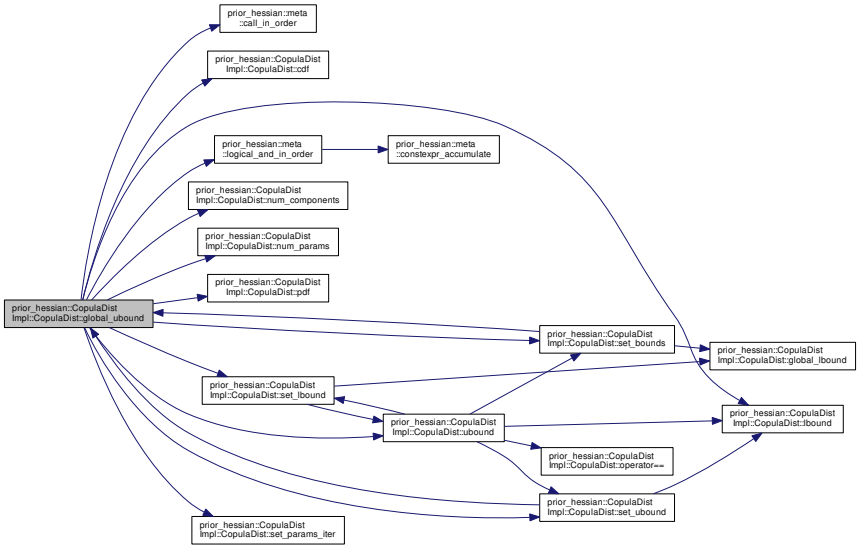
7.11.4.8 template<template< int > class CopulaTemplate, class... MarginalDistTs> const CopulaDist< CopulaTemplate, MarginalDistTs... >::NdimVecT & prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound ( ) [static]

Definition at line 545 of file CopulaDist.h.

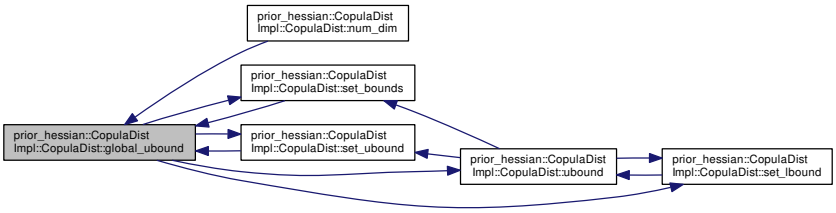
References prior_hessian::meta::call_in_order(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::cdf(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::lbound(), prior_hessian::meta::logical_and_in_order(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_components(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_params(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::pdf(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_bounds(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_lbound(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_params_iter(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::ubound().

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim(), prior_↩ hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_bounds(), and prior_hessian::Copula↩ DistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_ubound().

Here is the call graph for this function:



Here is the caller graph for this function:

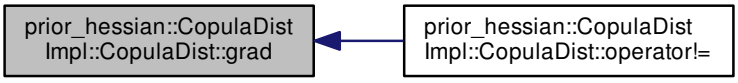


7.11.4.9 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > CopulaDist< CopulaTemplate, MarginalDistTs... >::NdimVecT prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::grad (const Vec & *x*) const

Definition at line 428 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

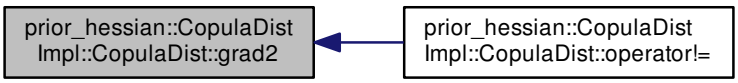


7.11.4.10 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > CopulaDist< CopulaTemplate, MarginalDistTs... >::NdimVecT prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::grad2 (const Vec & *x*) const

Definition at line 440 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

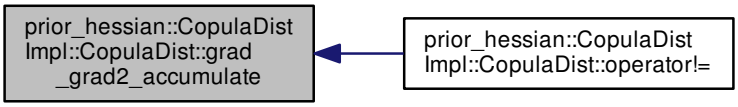


7.11.4.11 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec , class Vec2 > void prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::grad_grad2_accumulate (const Vec & *x*, Vec2 & *g*, Vec2 & *g2*) const

Definition at line 466 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

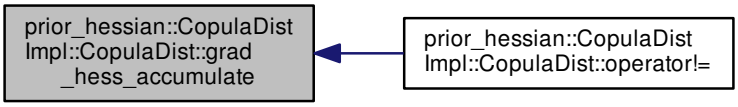


7.11.4.12 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec , class Vec2 , class Mat > void prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::grad_hess_accumulate (const Vec & *x,* Vec2 & *g,* Mat & *hess*) const

Definition at line 481 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

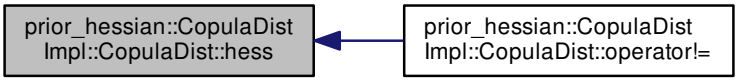


7.11.4.13 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > CopulaDist< CopulaTemplate, MarginalDistTs... >::NdimMatT prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::hess (const Vec & *x*) const

Definition at line 452 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

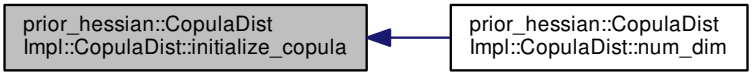


7.11.4.14 template<template< int > class CopulaTemplate, class... MarginalDistTs> void prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::initialize_copula (const CopulaT & *_copula*)

Definition at line 255 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim().

Here is the caller graph for this function:

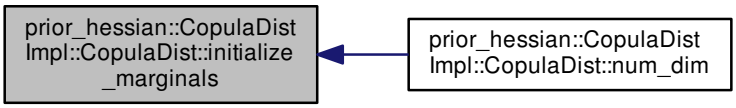


7.11.4.15 template<template< int > class CopulaTemplate, class... MarginalDistTs> void prior_hessian::CopulaDist↩ Impl::CopulaDist< CopulaTemplate, MarginalDistTs >::initialize_marginals (const MarginalDistTupleT & *dists*)

Definition at line 262 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim().

Here is the caller graph for this function:

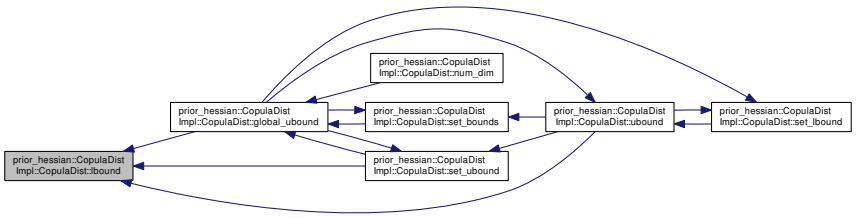


7.11.4.16 template<template< int > class CopulaTemplate, class... MarginalDistTs> NdimVecT prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::lbound ( ) const [inline]

Definition at line 67 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_ubound(), and prior_hessian::↩ CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::ubound().

Here is the caller graph for this function:

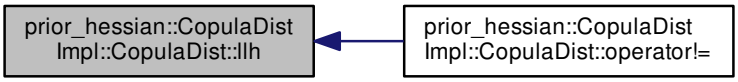


7.11.4.17 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > double prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::llh (const Vec & *x*) const

Definition at line 405 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

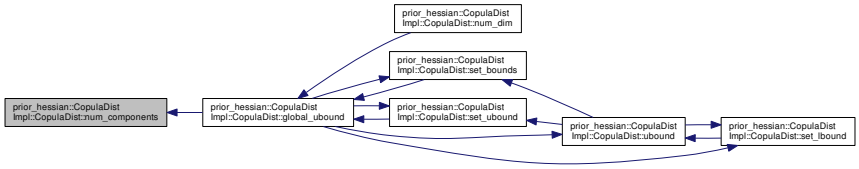


7.11.4.18 template<template< int > class CopulaTemplate, class... MarginalDistTs> static constexpr IdxT prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_components ( ) [inline], [static]

Definition at line 46 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound().

Here is the caller graph for this function:



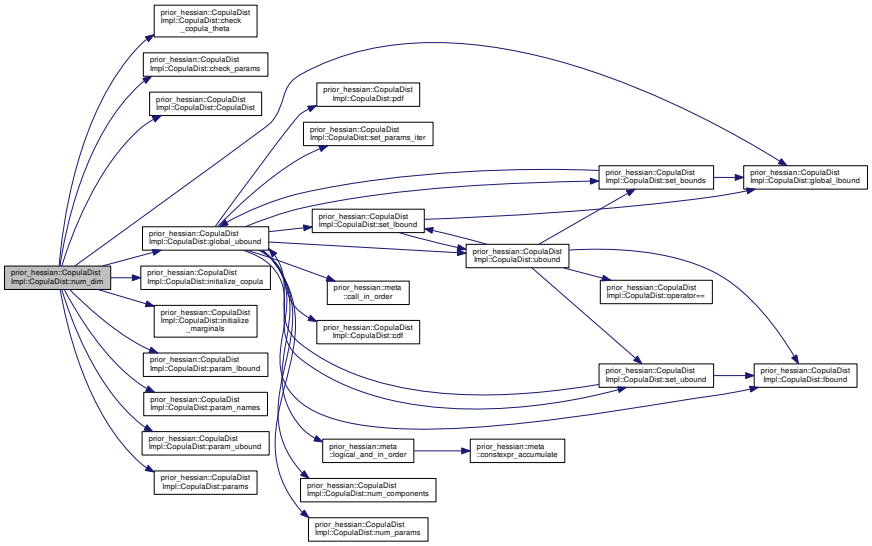
7.11.4.19 template<template< int > class CopulaTemplate, class... MarginalDistTs> static constexpr IdxT prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim ( ) [inline], [static]

Definition at line 47 of file CopulaDist.h.

References prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::check_copula_theta(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::check_params(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::CopulaDist(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_lbound(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate,

MarginalDistTs >::initialize_copula(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::initialize_marginals(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::param_↩ lbound(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::param_names(), prior_↩ hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::param_ubound(), and prior_hessian::↩ CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::params().

Here is the call graph for this function:

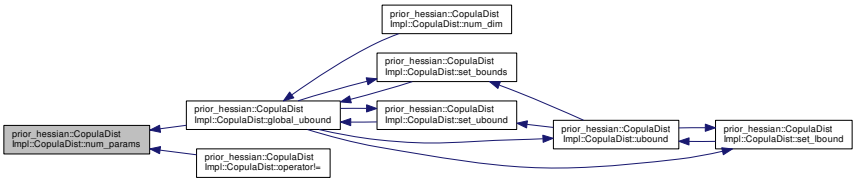


7.11.4.20 template<template< int > class CopulaTemplate, class... MarginalDistTs> static IdxT prior_hessian↩ ::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_params ( ) [inline], [static]

Definition at line 38 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

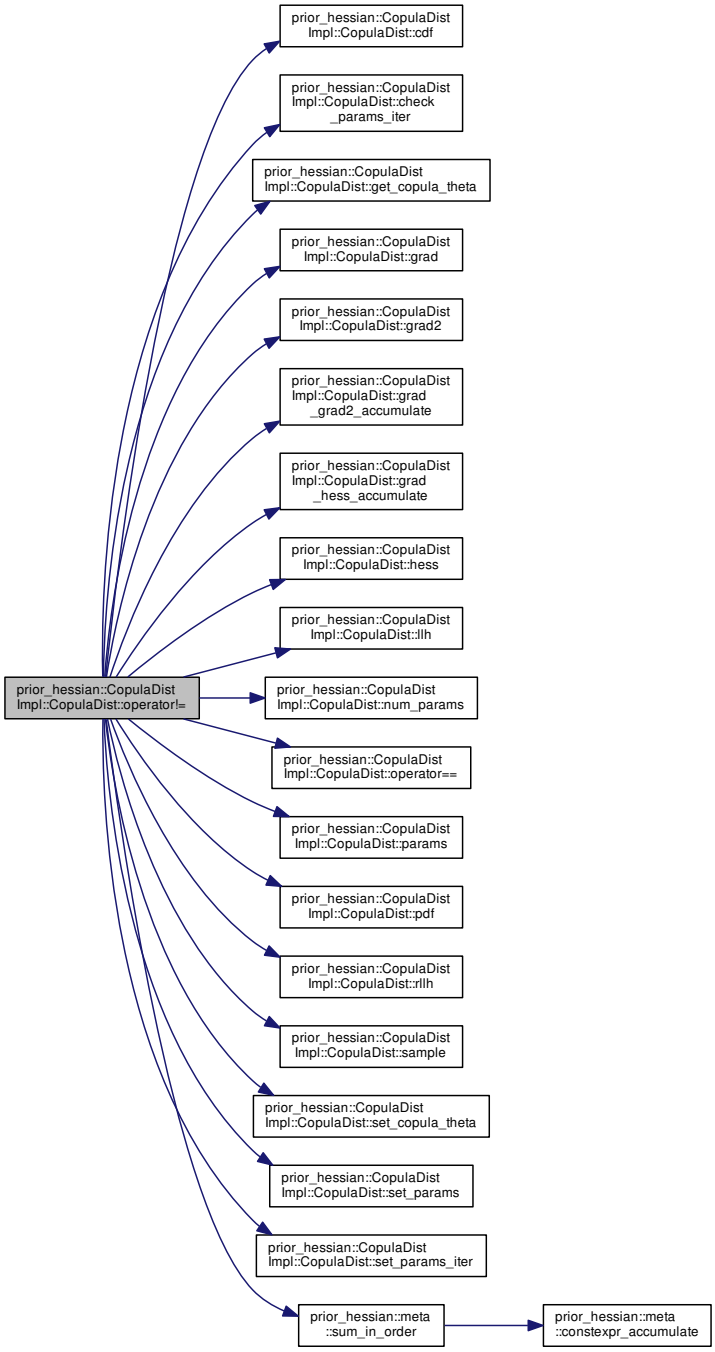


7.11.4.21 template<template< int > class CopulaTemplate, class... MarginalDistTs> bool prior_hessian::CopulaDist↩
Impl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!= (const CopulaDist< CopulaTemplate,
MarginalDistTs... > & *o*) const [inline]

Definition at line 80 of file CopulaDist.h.

References prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::cdf(), prior_hessian::↩
CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::check_params_iter(), prior_hessian::CopulaDist↩
Impl::CopulaDist< CopulaTemplate, MarginalDistTs >::get_copula_theta(), prior_hessian::CopulaDistImpl::Copula↩
Dist< CopulaTemplate, MarginalDistTs >::grad(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate,
MarginalDistTs >::grad2(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::grad_↩
grad2_accumulate(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::grad_hess_↩
accumulate(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::hess(), prior_hessian↩
::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::llh(), prior_hessian::CopulaDistImpl::CopulaDist<
CopulaTemplate, MarginalDistTs >::num_params(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate,
MarginalDistTs >::operator==(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >↩
::params(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::pdf(), prior_hessian↩
::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::rllh(), prior_hessian::CopulaDistImpl::Copula↩
Dist< CopulaTemplate, MarginalDistTs >::sample(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate,
MarginalDistTs >::set_copula_theta(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs
>::set_params(), prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_params_iter(),
and prior_hessian::meta::sum_in_order().

Here is the call graph for this function:

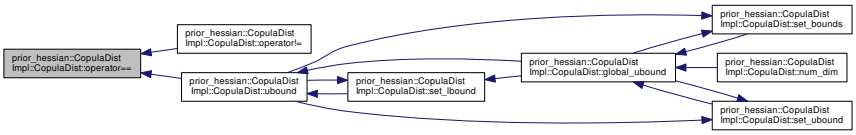


7.11.4.22 template<template< int > class CopulaTemplate, class... MarginalDistTs> bool prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator== (const CopulaDist< CopulaTemplate, MarginalDistTs... > & *o*) const

Definition at line 330 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::ubound().

Here is the caller graph for this function:

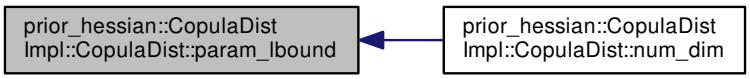


7.11.4.23 template<template< int > class CopulaTemplate, class... MarginalDistTs> const CopulaDist< CopulaTemplate, MarginalDistTs... >::NparamsVecT & prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::param_lbound ( ) [static]

Definition at line 518 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim().

Here is the caller graph for this function:

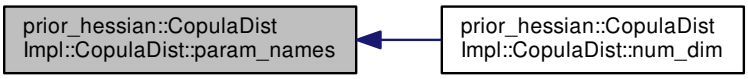


7.11.4.24 template<template< int > class CopulaTemplate, class... MarginalDistTs> const StringVecT & prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::param_names ( ) [static]

Definition at line 509 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim().

Here is the caller graph for this function:

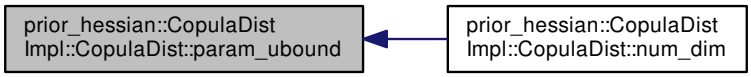


7.11.4.25 template<template< int > class CopulaTemplate, class... MarginalDistTs> const CopulaDist< CopulaTemplate, MarginalDistTs... >::NparamsVecT & prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::param_ubound ( ) [static]

Definition at line 527 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim().

Here is the caller graph for this function:

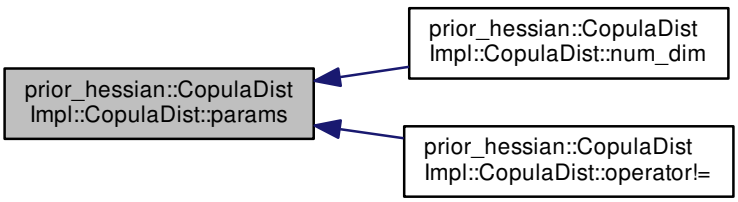


7.11.4.26 template<template< int > class CopulaTemplate, class... MarginalDistTs> CopulaDist< CopulaTemplate, MarginalDistTs... >::NparamsVecT prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::params ( ) const

Definition at line 337 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::num_dim(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

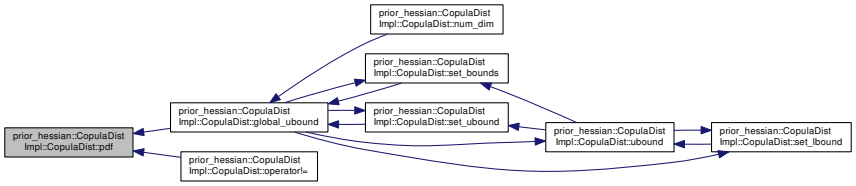


7.11.4.27 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > double prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::pdf (const Vec & *x*) const

Definition at line 394 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

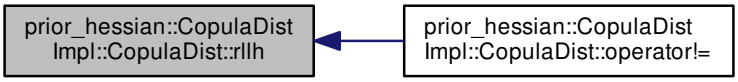


7.11.4.28 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > double prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::rllh (const Vec & *x*) const

Definition at line 416 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

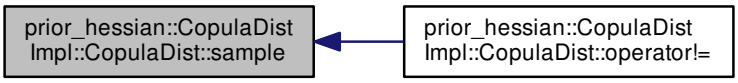


7.11.4.29 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class RngT > CopulaDist< CopulaTemplate, MarginalDistTs... >::NdimVecT prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::sample (RngT & *rng*) const

Definition at line 499 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:



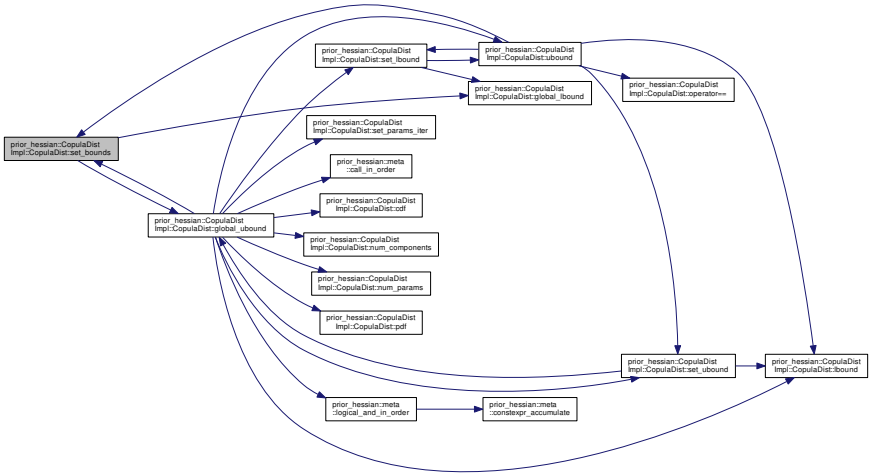
7.11.4.30 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec , class Vec2 > void prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_bounds (const Vec & *lbound,* const Vec2 & *ubound*)

Definition at line 271 of file CopulaDist.h.

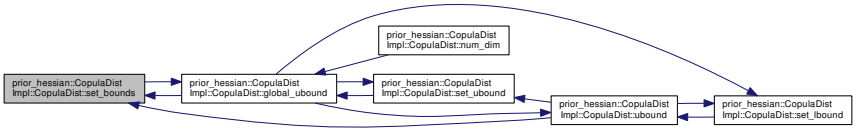
References prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_lbound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound().

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::ubound().

Here is the call graph for this function:



Here is the caller graph for this function:

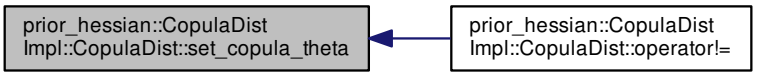


7.11.4.31 template<template< int > class CopulaTemplate, class... MarginalDistTs> void prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_copula_theta (double *theta*)

Definition at line 352 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:



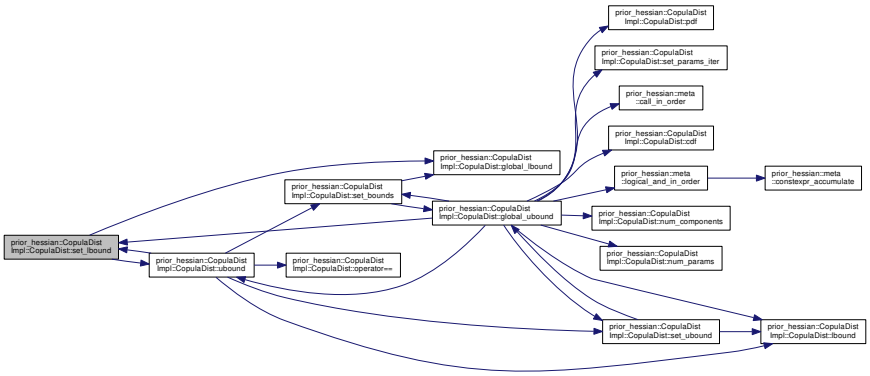
7.11.4.32 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > void prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_lbound (const Vec & *lbound*)

Definition at line 295 of file CopulaDist.h.

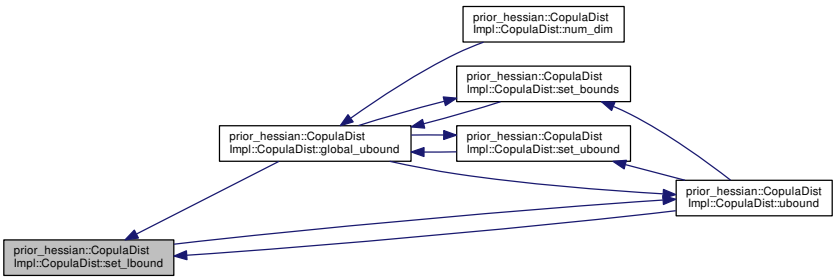
References prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_lbound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::ubound().

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::ubound().

Here is the call graph for this function:



Here is the caller graph for this function:

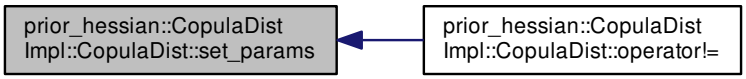


7.11.4.33 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > void prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_params (const Vec & *params*)

Definition at line 359 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:

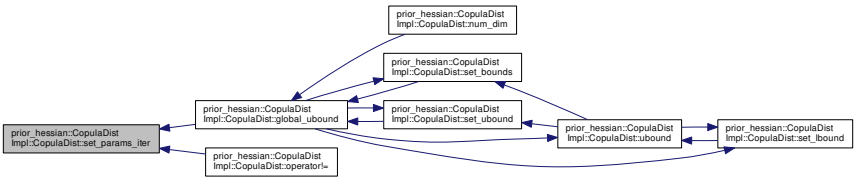


7.11.4.34 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class IterT > void prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_params_iter (IterT & *params*)

Definition at line 375 of file CopulaDist.h.

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator!=().

Here is the caller graph for this function:



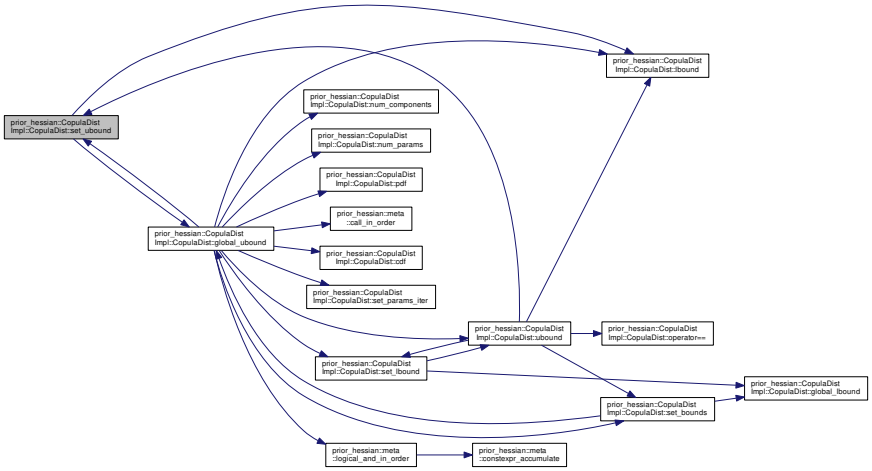
7.11.4.35 template<template< int > class CopulaTemplate, class... MarginalDistTs> template<class Vec > void prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_ubound (const Vec & *ubound*)

Definition at line 313 of file CopulaDist.h.

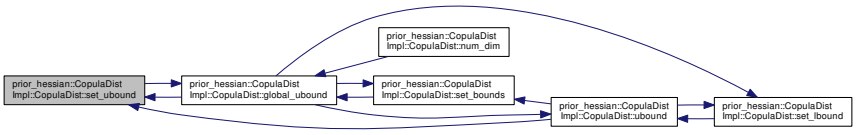
References prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::lbound().

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::ubound().

Here is the call graph for this function:



Here is the caller graph for this function:



7.11.4.36 template<template< int > class CopulaTemplate, class... MarginalDistTs> NdimVecT prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::ubound ( ) const [inline]

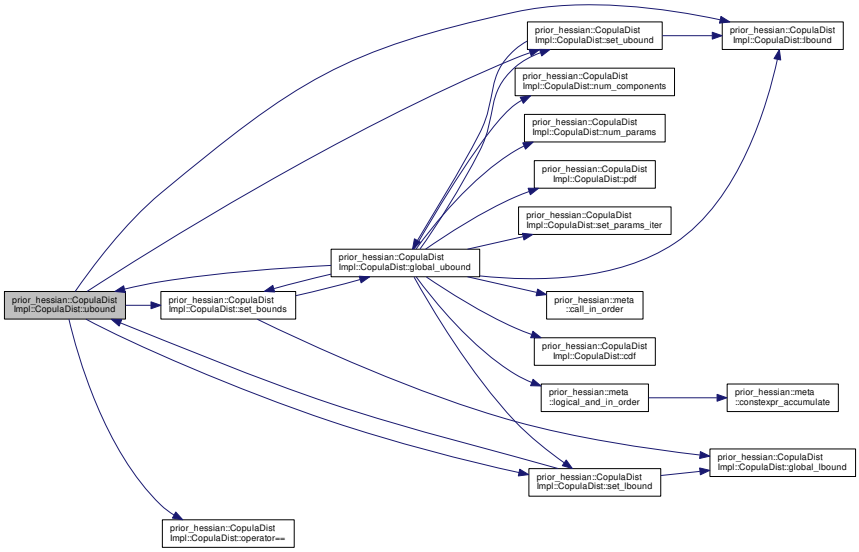
Definition at line 68 of file CopulaDist.h.

References prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::lbound(), prior_↩ hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::operator==(), prior_hessian::Copula↩ DistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_bounds(), prior_hessian::CopulaDistImpl::CopulaDist<

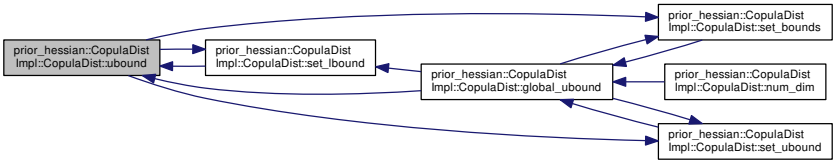
CopulaTemplate, MarginalDistTs >::set_lbound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_ubound().

Referenced by prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::global_ubound(), and prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >::set_lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



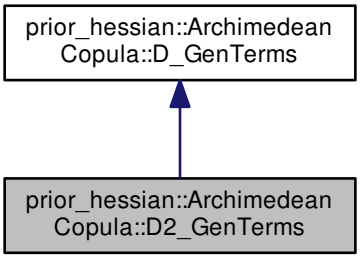
The documentation for this class was generated from the following file:

- CopulaDist.h

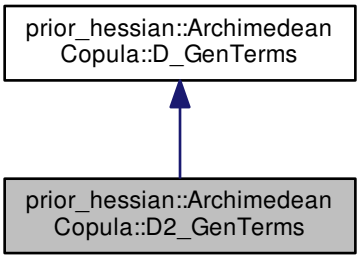
7.12 prior_hessian::ArchimedeanCopula::D2_GenTerms Struct Reference

```
#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩
ArchimedeanCopula.h>
```

Inheritance diagram for prior_hessian::ArchimedeanCopula::D2_GenTerms:



Collaboration diagram for prior_hessian::ArchimedeanCopula::D2_GenTerms:



Public Attributes

- double xi_n_t
- double log_dn_gen_t
- double eta_n_np1_t

7.12.1 Detailed Description

Definition at line 17 of file ArchimedeanCopula.h.

7.12.2 Member Data Documentation

7.12.2.1 double prior_hessian::ArchimedeanCopula::D_GenTerms::eta_n_np1_t [inherited]

Definition at line 15 of file ArchimedeanCopula.h.

7.12.2.2 double prior_hessian::ArchimedeanCopula::D_GenTerms::log_dn_gen_t [inherited]

Definition at line 14 of file ArchimedeanCopula.h.

7.12.2.3 double prior_hessian::ArchimedeanCopula::D2_GenTerms::xi_n_t

Definition at line 18 of file ArchimedeanCopula.h.

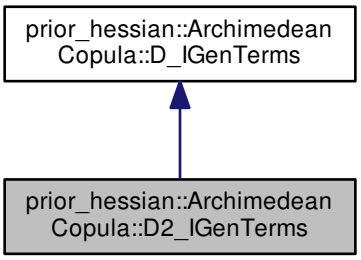
The documentation for this struct was generated from the following file:

- ArchimedeanCopula.h

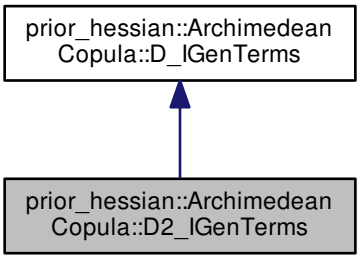
7.13 prior_hessian::ArchimedeanCopula::D2_lGenTerms Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ArchimedeanCopula.h>`

Inheritance diagram for prior_hessian::ArchimedeanCopula::D2_lGenTerms:



Collaboration diagram for prior_hessian::ArchimedeanCopula::D2_IGenTerms:



Public Attributes

- double d2_igen_ui
- double ixi_1_ui
- double d1_igen_ui
- double ieta_21_ui

7.13.1 Detailed Description

Definition at line 25 of file ArchimedeanCopula.h.

7.13.2 Member Data Documentation

7.13.2.1 double prior_hessian::ArchimedeanCopula::D_IGenTerms::d1_igen_ui [inherited]

Definition at line 22 of file ArchimedeanCopula.h.

7.13.2.2 double prior_hessian::ArchimedeanCopula::D2_IGenTerms::d2_igen_ui

Definition at line 26 of file ArchimedeanCopula.h.

7.13.2.3 double prior_hessian::ArchimedeanCopula::D_IGenTerms::ieta_21_ui [inherited]

Definition at line 23 of file ArchimedeanCopula.h.

7.13.2.4 double prior_hessian::ArchimedeanCopula::D2_IGenTerms::ixi_1_ui

Definition at line 27 of file ArchimedeanCopula.h.

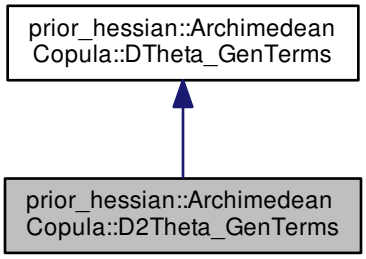
The documentation for this struct was generated from the following file:

- ArchimedeanCopula.h

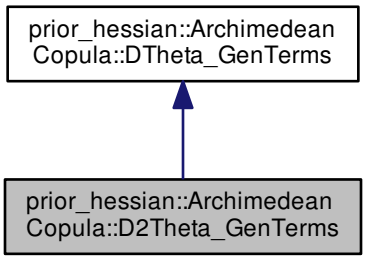
7.14 prior_hessian::ArchimedeanCopula::D2Theta_GenTerms Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ArchimedeanCopula.h>`

Inheritance diagram for prior_hessian::ArchimedeanCopula::D2Theta_GenTerms:



Collaboration diagram for prior_hessian::ArchimedeanCopula::D2Theta_GenTerms:



Public Attributes

- double xi_0n_t
- double log_dn_gen_t
- double eta_0n_1n_t

7.14.1 Detailed Description

Definition at line 34 of file ArchimedeanCopula.h.

7.14.2 Member Data Documentation

7.14.2.1 double prior_hessian::ArchimedeanCopula::DTheta_GenTerms::eta_0n_1n_t [inherited]

Definition at line 32 of file ArchimedeanCopula.h.

7.14.2.2 double prior_hessian::ArchimedeanCopula::DTheta_GenTerms::log_dn_gen_t [inherited]

Definition at line 31 of file ArchimedeanCopula.h.

7.14.2.3 double prior_hessian::ArchimedeanCopula::D2Theta_GenTerms::xi_0n_t

Definition at line 35 of file ArchimedeanCopula.h.

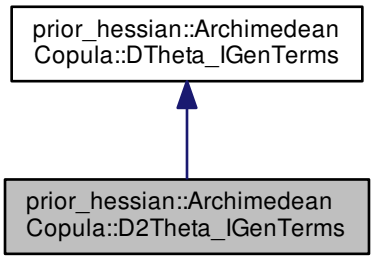
The documentation for this struct was generated from the following file:

- ArchimedeanCopula.h

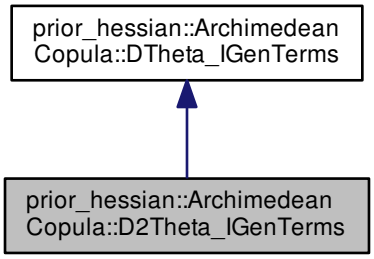
7.15 prior_hessian::ArchimedeanCopula::D2Theta_IGenTerms Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ArchimedeanCopula.h>`

Inheritance diagram for prior_hessian::ArchimedeanCopula::D2Theta_IGenTerms:



Collaboration diagram for prior_hessian::ArchimedeanCopula::D2Theta_IGenTerms:



Public Attributes

- double sum_d20_igen_u = 0
- double sum_ixi_01_u = 0
- double sum_log_d1_igen_u = 0
- double sum_d10_igen_u = 0
- double sum_ieta_01_11_u = 0

7.15.1 Detailed Description

Definition at line 43 of file ArchimedeanCopula.h.

7.15.2 Member Data Documentation

7.15.2.1 double prior_hessian::ArchimedeanCopula::DTheta_IGenTerms::sum_d10_igen_u = 0 [inherited]

Definition at line 40 of file ArchimedeanCopula.h.

7.15.2.2 double prior_hessian::ArchimedeanCopula::D2Theta_IGenTerms::sum_d20_igen_u = 0

Definition at line 44 of file ArchimedeanCopula.h.

7.15.2.3 double prior_hessian::ArchimedeanCopula::DTheta_IGenTerms::sum_ieta_01_11_u = 0 [inherited]

Definition at line 41 of file ArchimedeanCopula.h.

7.15.2.4 double prior_hessian::ArchimedeanCopula::D2Theta_IGenTerms::sum_ixi_01_u = 0

Definition at line 45 of file ArchimedeanCopula.h.

7.15.2.5 double prior_hessian::ArchimedeanCopula::DTheta_IGenTerms::sum_log_d1_igen_u = 0 [inherited]

Definition at line 39 of file ArchimedeanCopula.h.

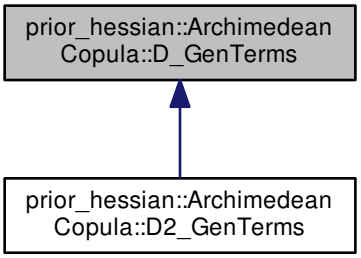
The documentation for this struct was generated from the following file:

- ArchimedeanCopula.h

7.16 prior_hessian::ArchimedeanCopula::D_GenTerms Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ArchimedeanCopula.h>`

Inheritance diagram for prior_hessian::ArchimedeanCopula::D_GenTerms:



Public Attributes

- double log_dn_gen_t
- double eta_n_np1_t

7.16.1 Detailed Description

Definition at line 13 of file ArchimedeanCopula.h.

7.16.2 Member Data Documentation

7.16.2.1 double prior_hessian::ArchimedeanCopula::D_GenTerms::eta_n_np1_t

Definition at line 15 of file ArchimedeanCopula.h.

7.16.2.2 double prior_hessian::ArchimedeanCopula::D_GenTerms::log_dn_gen_t

Definition at line 14 of file ArchimedeanCopula.h.

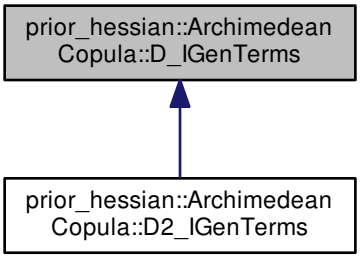
The documentation for this struct was generated from the following file:

- ArchimedeanCopula.h

7.17 prior_hessian::ArchimedeanCopula::D_lGenTerms Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ArchimedeanCopula.h>`

Inheritance diagram for prior_hessian::ArchimedeanCopula::D_lGenTerms:



Public Attributes

- double d1_igen_ui
- double ieta_21_ui

7.17.1 Detailed Description

Definition at line 21 of file ArchimedeanCopula.h.

7.17.2 Member Data Documentation

7.17.2.1 double prior_hessian::ArchimedeanCopula::D_lGenTerms::d1_igen_ui

Definition at line 22 of file ArchimedeanCopula.h.

7.17.2.2 double prior_hessian::ArchimedeanCopula::D_lGenTerms::ieta_21_ui

Definition at line 23 of file ArchimedeanCopula.h.

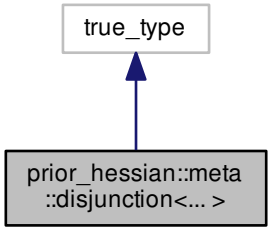
The documentation for this struct was generated from the following file:

- ArchimedeanCopula.h

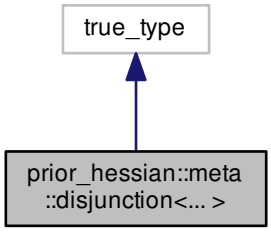
7.18 prior_hessian::meta::disjunction<... > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ Meta.h>`

Inheritance diagram for prior_hessian::meta::disjunction<... >:



Collaboration diagram for prior_hessian::meta::disjunction<... >:



7.18.1 Detailed Description

template<class...>
struct prior_hessian::meta::disjunction<... >

Definition at line 72 of file Meta.h.

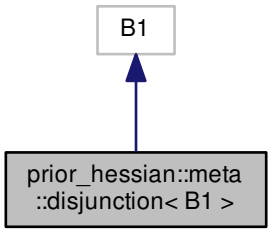
The documentation for this struct was generated from the following file:

- Meta.h

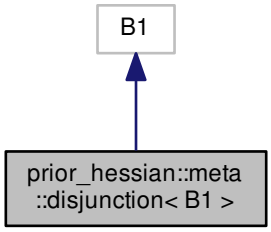
7.19 prior_hessian::meta::disjunction< B1 > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/Meta.h>`

Inheritance diagram for prior_hessian::meta::disjunction< B1 >:



Collaboration diagram for prior_hessian::meta::disjunction< B1 >:



7.19.1 Detailed Description

template<class B1>
struct prior_hessian::meta::disjunction< B1 >

Definition at line 73 of file Meta.h.

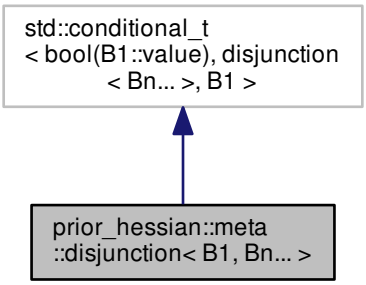
The documentation for this struct was generated from the following file:

- Meta.h

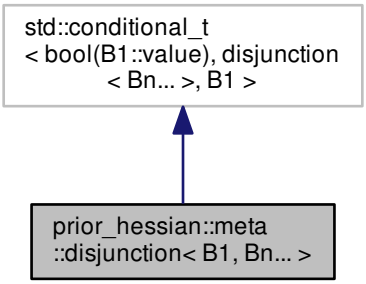
7.20 prior_hessian::meta::disjunction< B1, Bn... > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ Meta.h>`

Inheritance diagram for prior_hessian::meta::disjunction< B1, Bn... >:



Collaboration diagram for prior_hessian::meta::disjunction< B1, Bn... >:



7.20.1 Detailed Description

template<class B1, class... Bn>
struct prior_hessian::meta::disjunction< B1, Bn... >

Definition at line 75 of file Meta.h.

The documentation for this struct was generated from the following file:

- Meta.h

7.21 prior_hessian::detail::dist_adaptor_traits< Dist > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ BoundsAdaptedDist.h>`

Public Types

- using bounds_adapted_dist = void

Static Public Attributes

- static constexpr bool adaptable_bounds = false

7.21.1 Detailed Description

template<class Dist>
class prior_hessian::detail::dist_adaptor_traits< Dist >

Definition at line 42 of file BoundsAdaptedDist.h.

7.21.2 Member Typedef Documentation

7.21.2.1 template<class Dist> using prior_hessian::detail::dist_adaptor_traits< Dist >::bounds_adapted_dist = void

Definition at line 44 of file BoundsAdaptedDist.h.

7.21.3 Member Data Documentation

7.21.3.1 template<class Dist> constexpr bool prior_hessian::detail::dist_adaptor_traits< Dist >::adaptable_bounds = false [static]

Definition at line 45 of file BoundsAdaptedDist.h.

The documentation for this class was generated from the following file:

- BoundsAdaptedDist.h

7.22 prior_hessian::detail::dist_adaptor_traits< CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... > > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/CopulaDist.h>`

Public Types

- using bounds_adapted_dist = CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... >

Static Public Attributes

- static constexpr bool adaptable_bounds = true

7.22.1 Detailed Description

template<template< int > class CopulaTemplate, class... DistTs>
struct prior_hessian::detail::dist_adaptor_traits< CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... > >

Definition at line 198 of file CopulaDist.h.

7.22.2 Member Typedef Documentation

7.22.2.1 template<template< int > class CopulaTemplate, class... DistTs> using prior_hessian::detail::dist_adaptor_traits< CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... > >::bounds_adapted_dist = CopulaDistImpl::CopulaDist<CopulaTemplate, DistTs...>

Definition at line 200 of file CopulaDist.h.

7.22.3 Member Data Documentation

7.22.3.1 template<template< int > class CopulaTemplate, class... DistTs> constexpr bool prior_hessian::detail::dist_↩ adaptor_traits< CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... > >::adaptable_bounds = true [static]

Definition at line 201 of file CopulaDist.h.

The documentation for this struct was generated from the following file:

- CopulaDist.h

7.23 prior_hessian::detail::dist_adaptor_traits< GammaDist > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ TruncatedGammaDist.h>`

Public Types

- using bounds_adapted_dist = TruncatedGammaDist

Static Public Attributes

- static constexpr bool adaptable_bounds = false

7.23.1 Detailed Description

template<>
struct prior_hessian::detail::dist_adaptor_traits< GammaDist >

Definition at line 28 of file TruncatedGammaDist.h.

7.23.2 Member Typedef Documentation

7.23.2.1 using prior_hessian::detail::dist_adaptor_traits< GammaDist >::bounds_adapted_dist = TruncatedGammaDist

Definition at line 30 of file TruncatedGammaDist.h.

7.23.3 Member Data Documentation

7.23.3.1 constexpr bool prior_hessian::detail::dist_adaptor_traits< GammaDist >::adaptable_bounds = false [static]

Definition at line 31 of file TruncatedGammaDist.h.

The documentation for this struct was generated from the following file:

- TruncatedGammaDist.h

7.24 prior_hessian::detail::dist_adaptor_traits< MultivariateNormalDist< Ndim > > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ TruncatedMultivariateNormalDist.h>`

Public Types

- using bounds_adapted_dist = TruncatedMultivariateNormalDist< Ndim >

Static Public Attributes

- static constexpr bool adaptable_bounds = false

7.24.1 Detailed Description

template<IdxT Ndim>
struct prior_hessian::detail::dist_adaptor_traits< MultivariateNormalDist< Ndim > >

Definition at line 33 of file TruncatedMultivariateNormalDist.h.

7.24.2 Member Typedef Documentation

7.24.2.1 template<IdxT Ndim> using prior_hessian::detail::dist_adaptor_traits< MultivariateNormalDist< Ndim > >::bounds_adapted_dist = TruncatedMultivariateNormalDist<Ndim>

Definition at line 35 of file TruncatedMultivariateNormalDist.h.

7.24.3 Member Data Documentation

7.24.3.1 template<IdxT Ndim> constexpr bool prior_hessian::detail::dist_adaptor_traits< MultivariateNormalDist< Ndim > >::adaptable_bounds = false [static]

Definition at line 36 of file TruncatedMultivariateNormalDist.h.

The documentation for this struct was generated from the following file:

- TruncatedMultivariateNormalDist.h

7.25 prior_hessian::detail::dist_adaptor_traits< NormalDist > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ TruncatedNormalDist.h>`

Public Types

- using bounds_adapted_dist = TruncatedNormalDist

Static Public Attributes

- static constexpr bool adaptable_bounds = false

7.25.1 Detailed Description

template<>
struct prior_hessian::detail::dist_adaptor_traits< NormalDist >

Definition at line 30 of file TruncatedNormalDist.h.

7.25.2 Member Typedef Documentation

7.25.2.1 using prior_hessian::detail::dist_adaptor_traits< NormalDist >::bounds_adapted_dist = TruncatedNormalDist

Definition at line 32 of file TruncatedNormalDist.h.

7.25.3 Member Data Documentation

7.25.3.1 constexpr bool prior_hessian::detail::dist_adaptor_traits< NormalDist >::adaptable_bounds = false [static]

Definition at line 33 of file TruncatedNormalDist.h.

The documentation for this struct was generated from the following file:

- TruncatedNormalDist.h

7.26 prior_hessian::detail::dist_adaptor_traits< ParetoDist > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ TruncatedParetoDist.h>`

Public Types

- using bounds_adapted_dist = TruncatedParetoDist

Static Public Attributes

- static constexpr bool adaptable_bounds = false

7.26.1 Detailed Description

template<>
struct prior_hessian::detail::dist_adaptor_traits< ParetoDist >

Definition at line 28 of file TruncatedParetoDist.h.

7.26.2 Member Typedef Documentation

7.26.2.1 using prior_hessian::detail::dist_adaptor_traits< ParetoDist >::bounds_adapted_dist = TruncatedParetoDist

Definition at line 30 of file TruncatedParetoDist.h.

7.26.3 Member Data Documentation

7.26.3.1 constexpr bool prior_hessian::detail::dist_adaptor_traits< ParetoDist >::adaptable_bounds = false [static]

Definition at line 31 of file TruncatedParetoDist.h.

The documentation for this struct was generated from the following file:

- TruncatedParetoDist.h

7.27 prior_hessian::detail::dist_adaptor_traits< ScaledSymmetricBetaDist > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ScaledSymmetricBetaDist.h>`

Public Types

- using bounds_adapted_dist = ScaledSymmetricBetaDist

Static Public Attributes

- static constexpr bool adaptable_bounds = true

7.27.1 Detailed Description

template<>
struct prior_hessian::detail::dist_adaptor_traits< ScaledSymmetricBetaDist >

Definition at line 35 of file ScaledSymmetricBetaDist.h.

7.27.2 Member Typedef Documentation

7.27.2.1 using prior_hessian::detail::dist_adaptor_traits< ScaledSymmetricBetaDist >::bounds_adapted_dist = ScaledSymmetricBetaDist

Definition at line 37 of file ScaledSymmetricBetaDist.h.

7.27.3 Member Data Documentation

7.27.3.1 constexpr bool prior_hessian::detail::dist_adaptor_traits< ScaledSymmetricBetaDist >::adaptable_bounds = true [static]

Definition at line 38 of file ScaledSymmetricBetaDist.h.

The documentation for this struct was generated from the following file:

- ScaledSymmetricBetaDist.h

7.28 prior_hessian::detail::dist_adaptor_traits< SymmetricBetaDist > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ScaledSymmetricBetaDist.h>`

Public Types

- using bounds_adapted_dist = ScaledSymmetricBetaDist

Static Public Attributes

- static constexpr bool adaptable_bounds = false

7.28.1 Detailed Description

template<>
struct prior_hessian::detail::dist_adaptor_traits< SymmetricBetaDist >

Definition at line 28 of file ScaledSymmetricBetaDist.h.

7.28.2 Member Typedef Documentation

7.28.2.1 using prior_hessian::detail::dist_adaptor_traits< SymmetricBetaDist >::bounds_adapted_dist = ScaledSymmetricBetaDist

Definition at line 30 of file ScaledSymmetricBetaDist.h.

7.28.3 Member Data Documentation

7.28.3.1 constexpr bool prior_hessian::detail::dist_adaptor_traits< SymmetricBetaDist >::adaptable_bounds = false [static]

Definition at line 31 of file ScaledSymmetricBetaDist.h.

The documentation for this struct was generated from the following file:

- ScaledSymmetricBetaDist.h

7.29 prior_hessian::detail::dist_adaptor_traits< TruncatedGammaDist > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ TruncatedGammaDist.h>`

Public Types

- using bounds_adapted_dist = TruncatedGammaDist

Static Public Attributes

- static constexpr bool adaptable_bounds = true

7.29.1 Detailed Description

template<>
struct prior_hessian::detail::dist_adaptor_traits< TruncatedGammaDist >

Definition at line 35 of file TruncatedGammaDist.h.

7.29.2 Member Typedef Documentation

7.29.2.1 using prior_hessian::detail::dist_adaptor_traits< TruncatedGammaDist >::bounds_adapted_dist = TruncatedGammaDist

Definition at line 37 of file TruncatedGammaDist.h.

7.29.3 Member Data Documentation

7.29.3.1 constexpr bool prior_hessian::detail::dist_adaptor_traits< TruncatedGammaDist >::adaptable_bounds = true [static]

Definition at line 38 of file TruncatedGammaDist.h.

The documentation for this struct was generated from the following file:

- TruncatedGammaDist.h

7.30 prior_hessian::detail::dist_adaptor_traits< TruncatedMultivariateNormalDist< Ndim > > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ TruncatedMultivariateNormalDist.h>`

Public Types

- using bounds_adapted_dist = TruncatedMultivariateNormalDist< Ndim >

Static Public Attributes

- static constexpr bool adaptable_bounds = true

7.30.1 Detailed Description

template<IdxT Ndim>
struct prior_hessian::detail::dist_adaptor_traits< TruncatedMultivariateNormalDist< Ndim > >

Definition at line 40 of file TruncatedMultivariateNormalDist.h.

7.30.2 Member Typedef Documentation

7.30.2.1 template<IdxT Ndim> using prior_hessian::detail::dist_adaptor_traits< TruncatedMultivariateNormal↩ Dist< Ndim > >::bounds_adapted_dist = TruncatedMultivariateNormalDist<Ndim>

Definition at line 42 of file TruncatedMultivariateNormalDist.h.

7.30.3 Member Data Documentation

7.30.3.1 template<IdxT Ndim> constexpr bool prior_hessian::detail::dist_adaptor_traits< TruncatedMultivariateNormalDist< Ndim > >::adaptable_bounds = true [static]

Definition at line 43 of file TruncatedMultivariateNormalDist.h.

The documentation for this struct was generated from the following file:

- TruncatedMultivariateNormalDist.h

7.31 prior_hessian::detail::dist_adaptor_traits< TruncatedNormalDist > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ TruncatedNormalDist.h>`

Public Types

- using bounds_adapted_dist = TruncatedNormalDist

Static Public Attributes

- static constexpr bool adaptable_bounds = true

7.31.1 Detailed Description

template<>
struct prior_hessian::detail::dist_adaptor_traits< TruncatedNormalDist >

Definition at line 37 of file TruncatedNormalDist.h.

7.31.2 Member Typedef Documentation

7.31.2.1 using prior_hessian::detail::dist_adaptor_traits< TruncatedNormalDist >::bounds_adapted_dist = TruncatedNormalDist

Definition at line 39 of file TruncatedNormalDist.h.

7.31.3 Member Data Documentation

7.31.3.1 constexpr bool prior_hessian::detail::dist_adaptor_traits< TruncatedNormalDist >::adaptable_bounds = true [static]

Definition at line 40 of file TruncatedNormalDist.h.

The documentation for this struct was generated from the following file:

- TruncatedNormalDist.h

7.32 prior_hessian::detail::dist_adaptor_traits< TruncatedParetoDist > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ TruncatedParetoDist.h>`

Public Types

- using bounds_adapted_dist = TruncatedParetoDist

Static Public Attributes

- static constexpr bool adaptable_bounds = true

7.32.1 Detailed Description

template<>
struct prior_hessian::detail::dist_adaptor_traits< TruncatedParetoDist >

Definition at line 35 of file TruncatedParetoDist.h.

7.32.2 Member Typedef Documentation

7.32.2.1 using prior_hessian::detail::dist_adaptor_traits< TruncatedParetoDist >::bounds_adapted_dist = TruncatedParetoDist

Definition at line 37 of file TruncatedParetoDist.h.

7.32.3 Member Data Documentation

7.32.3.1 constexpr bool prior_hessian::detail::dist_adaptor_traits< TruncatedParetoDist >::adaptable_bounds = true [static]

Definition at line 38 of file TruncatedParetoDist.h.

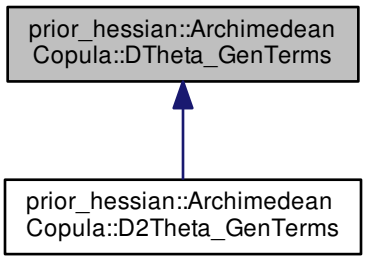
The documentation for this struct was generated from the following file:

- TruncatedParetoDist.h

7.33 prior_hessian::ArchimedeanCopula::DTheta_GenTerms Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ArchimedeanCopula.h>`

Inheritance diagram for prior_hessian::ArchimedeanCopula::DTheta_GenTerms:



Public Attributes

- double log_dn_gen_t
- double eta_0n_1n_t

7.33.1 Detailed Description

Definition at line 30 of file ArchimedeanCopula.h.

7.33.2 Member Data Documentation

7.33.2.1 double prior_hessian::ArchimedeanCopula::DTheta_GenTerms::eta_0n_1n_t

Definition at line 32 of file ArchimedeanCopula.h.

7.33.2.2 double prior_hessian::ArchimedeanCopula::DTheta_GenTerms::log_dn_gen_t

Definition at line 31 of file ArchimedeanCopula.h.

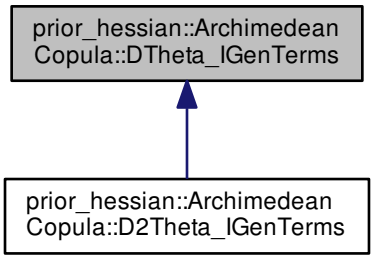
The documentation for this struct was generated from the following file:

- ArchimedeanCopula.h

7.34 prior_hessian::ArchimedeanCopula::DTheta_IGenTerms Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ ArchimedeanCopula.h>`

Inheritance diagram for prior_hessian::ArchimedeanCopula::DTheta_IGenTerms:



Public Attributes

- double sum_log_d1_igen_u = 0
- double sum_d10_igen_u = 0
- double sum_ieta_01_11_u = 0

7.34.1 Detailed Description

Definition at line 38 of file ArchimedeanCopula.h.

7.34.2 Member Data Documentation

7.34.2.1 double prior_hessian::ArchimedeanCopula::DTheta_IGenTerms::sum_d10_igen_u = 0

Definition at line 40 of file ArchimedeanCopula.h.

7.34.2.2 double prior_hessian::ArchimedeanCopula::DTheta_lGenTerms::sum_ieta_01_11_u = 0

Definition at line 41 of file ArchimedeanCopula.h.

7.34.2.3 double prior_hessian::ArchimedeanCopula::DTheta_lGenTerms::sum_log_d1_igen_u = 0

Definition at line 39 of file ArchimedeanCopula.h.

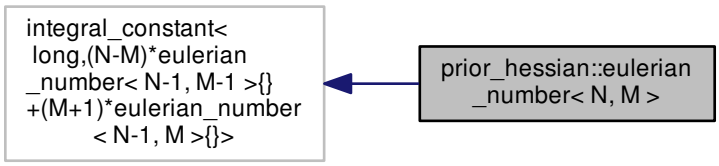
The documentation for this struct was generated from the following file:

- ArchimedeanCopula.h

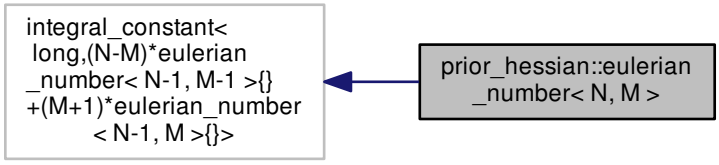
7.35 prior_hessian::eulerian_number< N, M > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ EulerianPolynomial.h>`

Inheritance diagram for prior_hessian::eulerian_number< N, M >:



Collaboration diagram for prior_hessian::eulerian_number< N, M >:



7.35.1 Detailed Description

template<long N, long M>
struct prior_hessian::eulerian_number< N, M >

Definition at line 17 of file EulerianPolynomial.h.

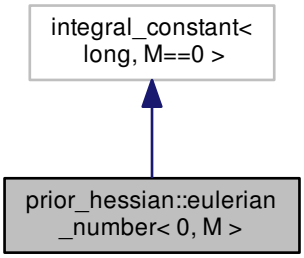
The documentation for this struct was generated from the following file:

- EulerianPolynomial.h

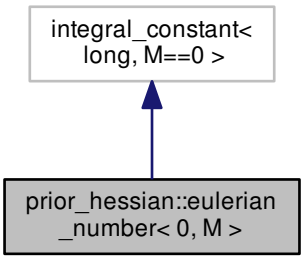
7.36 prior_hessian::eulerian_number< 0, M > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ EulerianPolynomial.h>`

Inheritance diagram for prior_hessian::eulerian_number< 0, M >:



Collaboration diagram for prior_hessian::eulerian_number< 0, M >:



7.36.1 Detailed Description

template<long M>
struct prior_hessian::eulerian_number< 0, M >

Definition at line 20 of file EulerianPolynomial.h.

The documentation for this struct was generated from the following file:

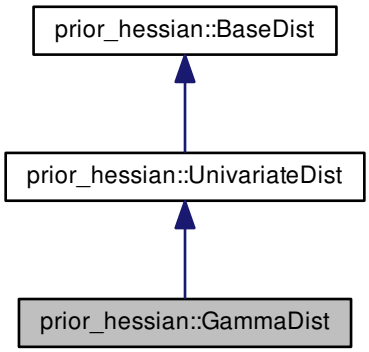
- EulerianPolynomial.h

7.37 prior_hessian::GammaDist Class Reference

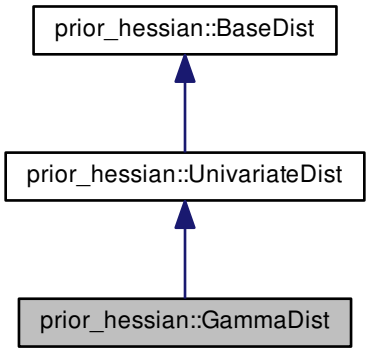
Gamma distribution.

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/GammaDist.h>`

Inheritance diagram for prior_hessian::GammaDist:



Collaboration diagram for prior_hessian::GammaDist:



Public Types

- using NparamsVecT = arma::Col< double >::fixed< _num_params >

Public Member Functions

- GammaDist (double scale, double shape)
- GammaDist ()
- template<class Vec , meta::ConstructableIfNotSelfT< Vec, GammaDist > = true>
  GammaDist (const Vec &params)
- double get_param (int idx) const
- void set_param (int idx, double val)
- NparamsVecT params () const
- template<class Vec >
  void set_params (const Vec &p)
- void set_params (double scale, double shape)
- bool operator== (const GammaDist &o) const
- bool operator!= (const GammaDist &o) const
- double scale () const
- double shape () const
- void set_scale (double val)
- void set_shape (double val)
- double mean () const
- double median () const
- double cdf (double x) const
- double icdf (double u) const
- double pdf (double x) const
- double llh (double x) const
- double rllh (double x) const

- double grad (double x) const
- double grad2 (double x) const
- void grad_grad2_accumulate (double x, double &g, double &g2) const
- template<class RngT >
  double sample (RngT &rng) const
- template<class IterT >
  void set_params_iter (IterT &params)

Static Public Member Functions

- static constexpr IdxT num_params ()
- static constexpr double lbound ()
- static constexpr double ubound ()
- static bool in_bounds (double u)
- static const StringVecT & param_names ()
- static const NparamsVecT & param_lbound ()
- static const NparamsVecT & param_ubound ()
- static bool check_params (double shape, double scale)
- template<class Vec >
  static bool check_params (const Vec &params)
- template<class IterT >
  static bool check_params_iter (IterT &params)
- static constexpr IdxT num_dim ()

Static Protected Member Functions

- static void check_bounds (double lbound, double ubound)

7.37.1 Detailed Description

Gamma distribution.

Definition at line 22 of file GammaDist.h.

7.37.2 Member Typedef Documentation

7.37.2.1 using prior_hessian::GammaDist::NparamsVecT = arma::Col<double>::fixed<_num_params>

Definition at line 26 of file GammaDist.h.

7.37.3 Constructor & Destructor Documentation

7.37.3.1 prior_hessian::GammaDist::GammaDist (double *scale,* double *shape*)

Definition at line 25 of file GammaDist.cpp.

7.37.3.2 prior_hessian::GammaDist::GammaDist ( ) [inline]

Definition at line 43 of file GammaDist.h.

Referenced by param_ubound().

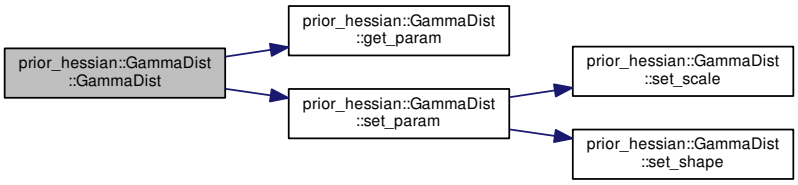
Here is the caller graph for this function:

7.37.3.3 template<class Vec , meta::ConstructableIfNotSelfT< Vec, GammaDist > = true> prior_hessian::GammaDist::GammaDist (const Vec & *params*) [inline], [explicit]

Definition at line 45 of file GammaDist.h.

References get_param(), and set_param().

Here is the call graph for this function:



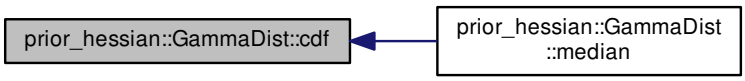
7.37.4 Member Function Documentation

7.37.4.1 double prior_hessian::GammaDist::cdf (double *x*) const

Definition at line 53 of file GammaDist.cpp.

Referenced by median().

Here is the caller graph for this function:



7.37.4.2 void prior_hessian::UnivariateDist::check_bounds (double *lbound,* double *ubound*) [static],[protected], [inherited]

Definition at line 17 of file UnivariateDist.cpp.

Referenced by prior_hessian::UnivariateDist::UnivariateDist().

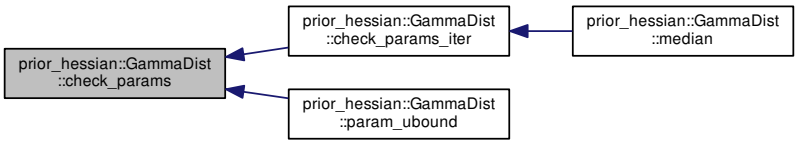
Here is the caller graph for this function:

7.37.4.3 bool prior_hessian::GammaDist::check_params (double *shape,* double *scale*) [inline],[static]

Definition at line 107 of file GammaDist.h.

Referenced by check_params_iter(), and param_ubound().

Here is the caller graph for this function:



7.37.4.4 template<class Vec > bool prior_hessian::GammaDist::check_params (const Vec & *params*) [static]

Definition at line 113 of file GammaDist.h.

References params().

Here is the call graph for this function:

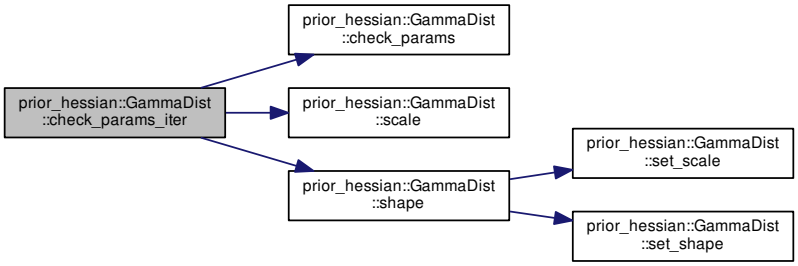
7.37.4.5 template<class IterT > bool prior_hessian::GammaDist::check_params_iter (IterT & *params*) [static]

Definition at line 183 of file GammaDist.h.

References check_params(), scale(), and shape().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

7.37.4.6 double prior_hessian::GammaDist::get_param (int *idx*) const [inline]

Definition at line 119 of file GammaDist.h.

Referenced by GammaDist().

Here is the caller graph for this function:

7.37.4.7 double prior_hessian::GammaDist::grad (double *x*) const [inline]

Definition at line 155 of file GammaDist.h.

Referenced by median().

Here is the caller graph for this function:

7.37.4.8 double prior_hessian::GammaDist::grad2 (double *x*) const [inline]

Definition at line 161 of file GammaDist.h.

Referenced by median().

Here is the caller graph for this function:

7.37.4.9 void prior_hessian::GammaDist::grad_grad2_accumulate (double *x*, double & *g*, double & *g2*) const [inline]

Definition at line 167 of file GammaDist.h.

Referenced by median().

Here is the caller graph for this function:

7.37.4.10 double prior_hessian::GammaDist::icdf (double *u*) const

Definition at line 58 of file GammaDist.cpp.

Referenced by median().

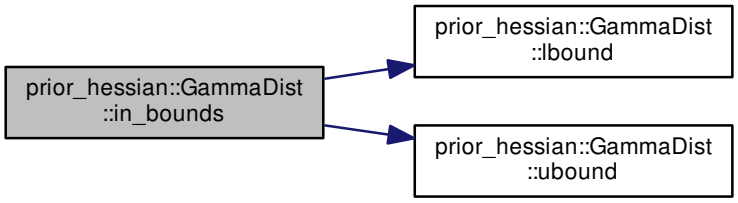
Here is the caller graph for this function:

7.37.4.11 static bool prior_hessian::GammaDist::in_bounds (double *u*) [inline], [static]

Definition at line 32 of file GammaDist.h.

References lbound(), and ubound().

Here is the call graph for this function:



7.37.4.12 static constexpr double prior_hessian::GammaDist::lbound ( ) [inline],[static]

Definition at line 30 of file GammaDist.h.

Referenced by in_bounds().

Here is the caller graph for this function:

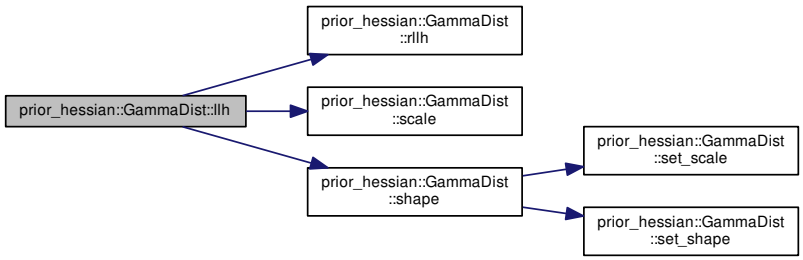
7.37.4.13 double prior_hessian::GammaDist::llh (double *x*) const

Definition at line 72 of file GammaDist.cpp.

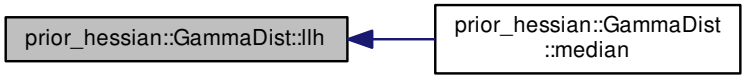
References rllh(), scale(), and shape().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:



7.37.4.14 double prior_hessian::GammaDist::mean ( ) const [inline]

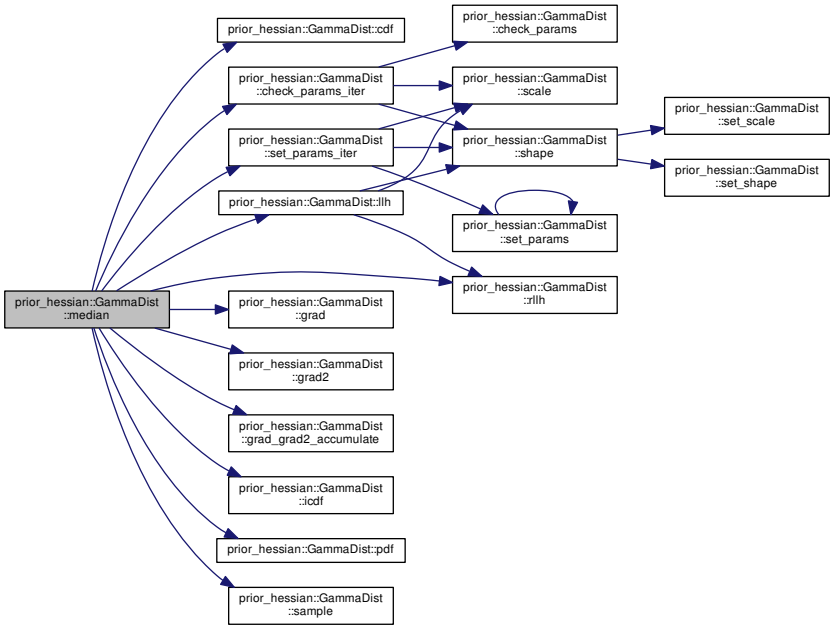
Definition at line 61 of file GammaDist.h.

7.37.4.15 double prior_hessian::GammaDist::median ( ) const [inline]

Definition at line 62 of file GammaDist.h.

References cdf(), check_params_iter(), grad(), grad2(), grad_grad2_accumulate(), icdf(), llh(), pdf(), rllh(), sample(), and set_params_iter().

Here is the call graph for this function:



7.37.4.16 static constexpr IdxT prior_hessian::UnivariateDist::num_dim ( ) [inline], [static], [inherited]

Definition at line 17 of file UnivariateDist.h.

7.37.4.17 static constexpr IdxT prior_hessian::GammaDist::num_params ( ) [inline], [static]

Definition at line 29 of file GammaDist.h.

7.37.4.18 bool prior_hessian::GammaDist::operator!= (const GammaDist & *o*) const [inline]

Definition at line 54 of file GammaDist.h.

References operator==().

Here is the call graph for this function:

7.37.4.19 bool prior_hessian::GammaDist::operator== (const GammaDist & *o*) const [inline]

Definition at line 53 of file GammaDist.h.

Referenced by operator!=().

Here is the caller graph for this function:

7.37.4.20 static const NparamsVecT& prior_hessian::GammaDist::param_lbound ( ) [inline], [static]

Definition at line 35 of file GammaDist.h.

7.37.4.21 static const StringVecT& prior_hessian::GammaDist::param_names ( ) [inline], [static]

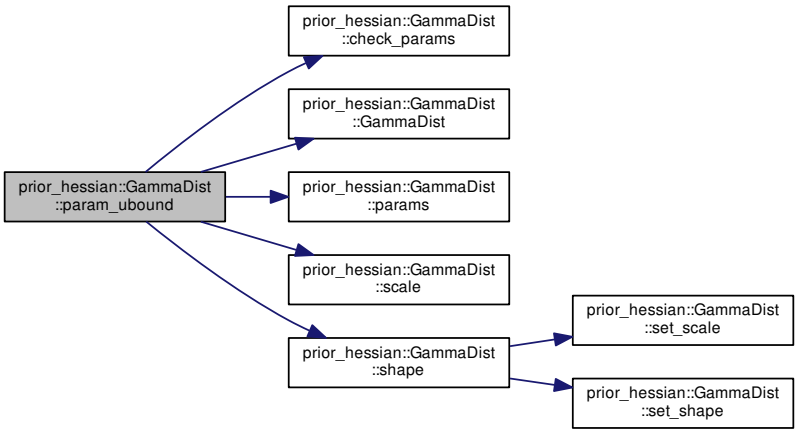
Definition at line 34 of file GammaDist.h.

7.37.4.22 static const NparamsVecT& prior_hessian::GammaDist::param_ubound ( ) [inline], [static]

Definition at line 36 of file GammaDist.h.

References check_params(), GammaDist(), params(), scale(), and shape().

Here is the call graph for this function:

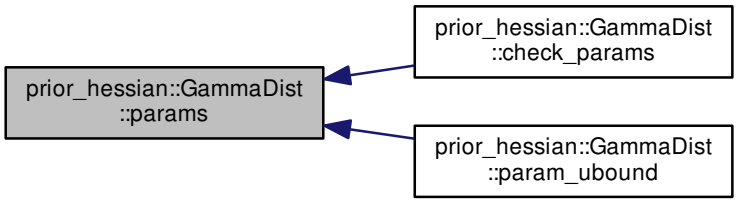


7.37.4.23 NparamsVecT prior_hessian::GammaDist::params ( ) const [inline]

Definition at line 49 of file GammaDist.h.

Referenced by check_params(), and param_ubound().

Here is the caller graph for this function:

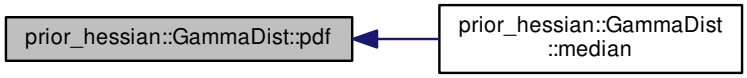


7.37.4.24 double prior_hessian::GammaDist::pdf (double *x*) const

Definition at line 65 of file GammaDist.cpp.

Referenced by median().

Here is the caller graph for this function:



7.37.4.25 double prior_hessian::GammaDist::rllh (double *x*) const [inline]

Definition at line 149 of file GammaDist.h.

Referenced by llh(), and median().

Here is the caller graph for this function:

7.37.4.26 template<class RngT > double prior_hessian::GammaDist::sample (RngT & *rng*) const

Definition at line 175 of file GammaDist.h.

Referenced by median().

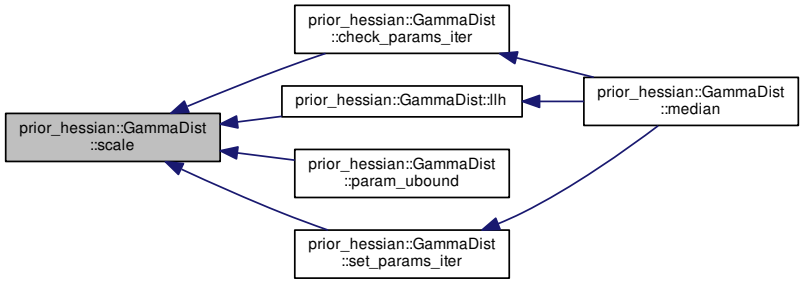
Here is the caller graph for this function:

7.37.4.27 double prior_hessian::GammaDist::scale ( ) const [inline]

Definition at line 56 of file GammaDist.h.

Referenced by check_params_iter(), llh(), param_ubound(), and set_params_iter().

Here is the caller graph for this function:



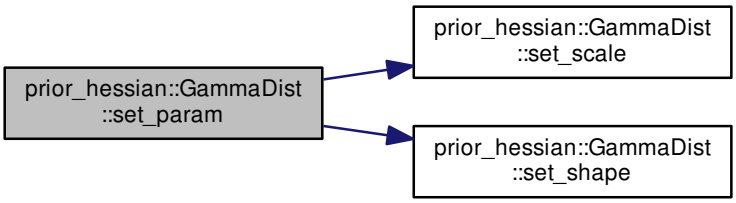
7.37.4.28 void prior_hessian::GammaDist::set_param (int *idx*, double *val*) [inline]

Definition at line 133 of file GammaDist.h.

References set_scale(), and set_shape().

Referenced by GammaDist().

Here is the call graph for this function:



Here is the caller graph for this function:

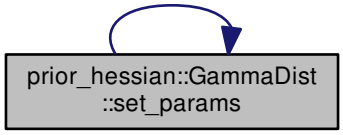
7.37.4.29 template<class Vec > void prior_hessian::GammaDist::set_params (const Vec & *p*) [inline]

Definition at line 51 of file GammaDist.h.

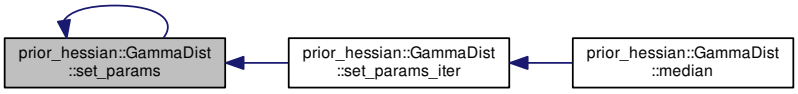
References set_params().

Referenced by set_params(), and set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:



7.37.4.30 void prior_hessian::GammaDist::set_params (double *scale*, double *shape*)

Definition at line 46 of file GammaDist.cpp.

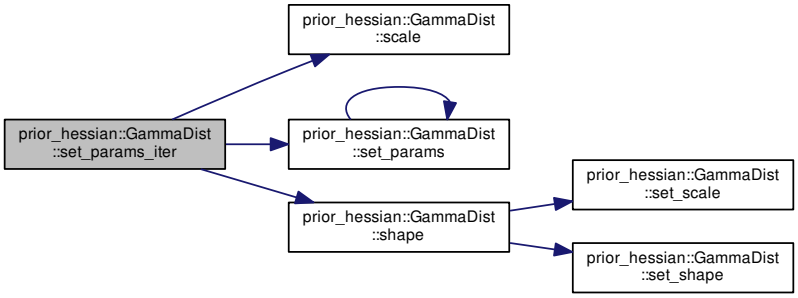
7.37.4.31 template<class IterT > void prior_hessian::GammaDist::set_params_iter (IterT & *params*)

Definition at line 191 of file GammaDist.h.

References scale(), set_params(), and shape().

Referenced by median().

Here is the call graph for this function:



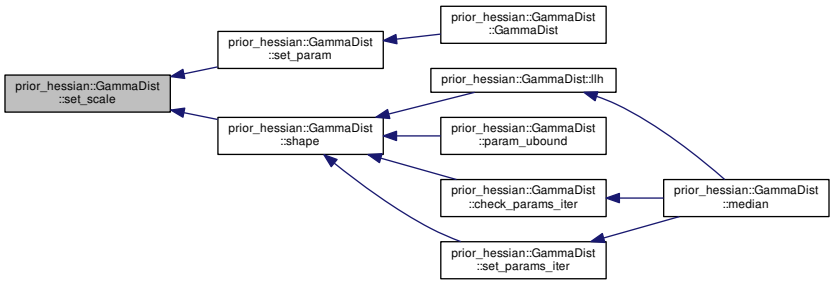
Here is the caller graph for this function:

7.37.4.32 void prior_hessian::GammaDist::set_scale (double *val*)

Definition at line 34 of file GammaDist.cpp.

Referenced by set_param(), and shape().

Here is the caller graph for this function:

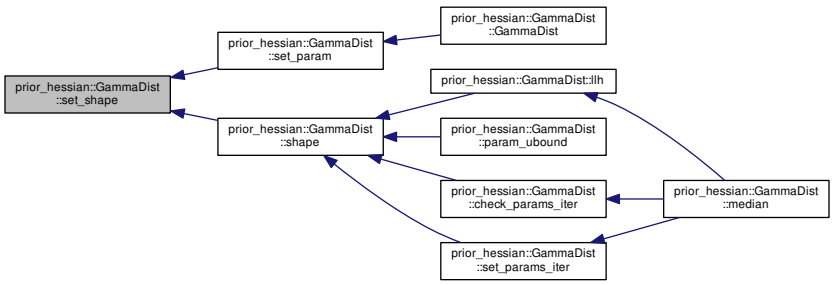


7.37.4.33 void prior_hessian::GammaDist::set_shape (double *val*)

Definition at line 40 of file GammaDist.cpp.

Referenced by set_param(), and shape().

Here is the caller graph for this function:



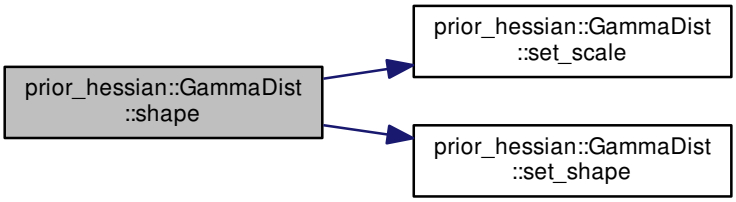
7.37.4.34 double prior_hessian::GammaDist::shape ( ) const [inline]

Definition at line 57 of file GammaDist.h.

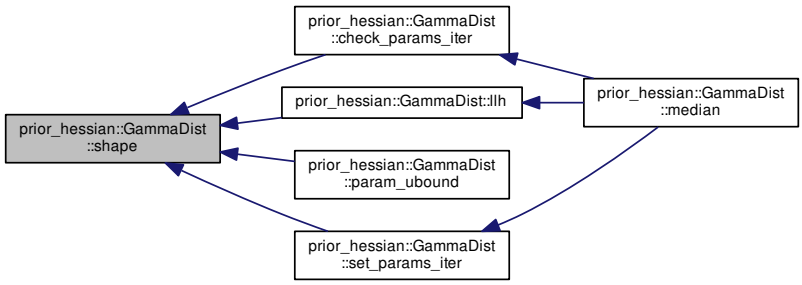
References set_scale(), and set_shape().

Referenced by check_params_iter(), llh(), param_ubound(), and set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:



7.37.4.35 static constexpr double prior_hessian::GammaDist::ubound ( ) [inline], [static]

Definition at line 31 of file GammaDist.h.

Referenced by in_bounds().

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

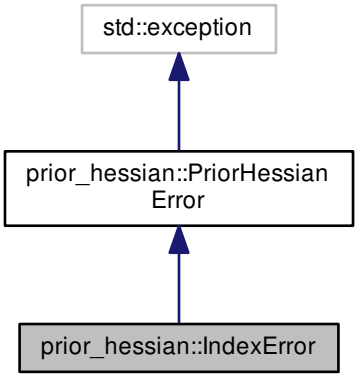
- GammaDist.h
- GammaDist.cpp

7.38 prior_hessian::IndexError Struct Reference

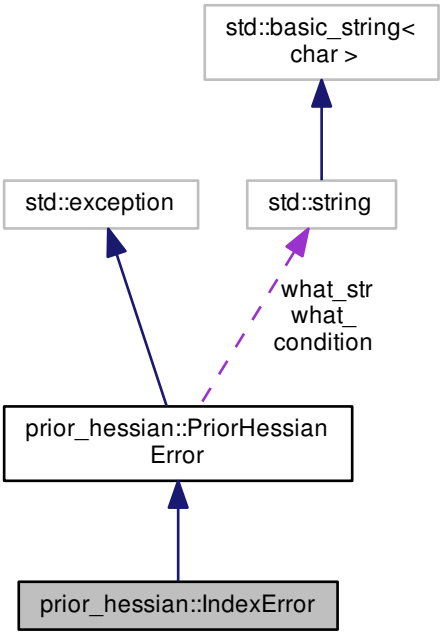
Indicates a index access was out of bounds.

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::IndexError:



Collaboration diagram for prior_hessian::IndexError:



Public Member Functions

- IndexError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.38.1 Detailed Description

Indicates a index access was out of bounds.

Definition at line 65 of file PriorHessianError.h.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 prior_hessian::IndexError::IndexError (std::string *message*) [inline]

Definition at line 67 of file PriorHessianError.h.

7.38.3 Member Function Documentation

7.38.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.38.4 Member Data Documentation

7.38.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.38.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.38.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

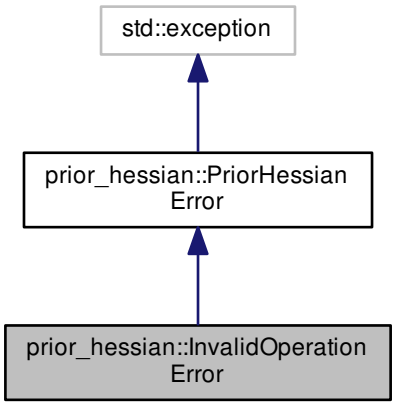
The documentation for this struct was generated from the following file:

- PriorHessianError.h

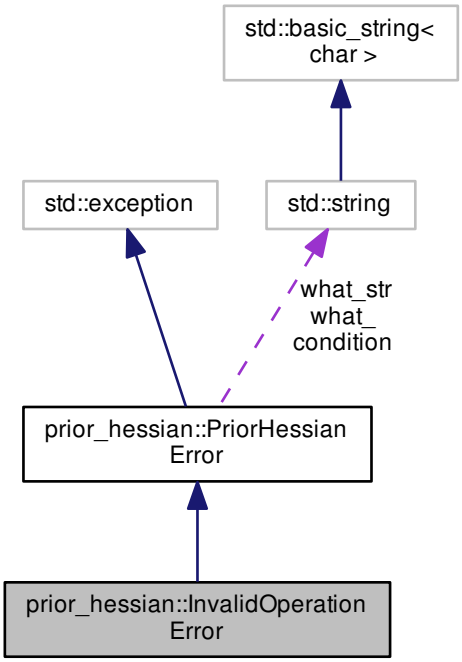
7.39 prior_hessian::InvalidOperationError Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::InvalidOperationError:



Collaboration diagram for prior_hessian::InvalidOperationError:



Public Member Functions

- InvalidOperationError (std::string message)
- const char ∗ what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.39.1 Detailed Description

Definition at line 70 of file PriorHessianError.h.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 prior_hessian::InvalidOperationError::InvalidOperationError (std::string *message*) [inline]

Definition at line 72 of file PriorHessianError.h.

7.39.3 Member Function Documentation

7.39.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.39.4 Member Data Documentation

7.39.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.39.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.39.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

The documentation for this struct was generated from the following file:

- PriorHessianError.h

7.40 prior_hessian::meta::is_copula< T, U > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/Meta.h>`

Static Public Attributes

- static constexpr bool value = decltype(is_copula::test(std::declval<U>()))::value

7.40.1 Detailed Description

template<template< template< int > class, class... > class T, typename U>
class prior_hessian::meta::is_copula< T, U >

Definition at line 102 of file Meta.h.

7.40.2 Member Data Documentation

7.40.2.1 template<template< template< int > class, class... > class T, typename U > constexpr bool prior_hessian::meta::is_copula< T, U >::value = decltype(is_copula::test(std::declval<U>()))::value [static]

Definition at line 107 of file Meta.h.

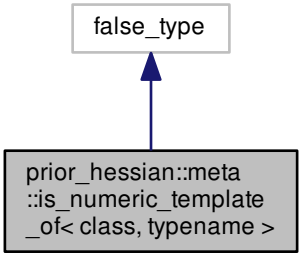
The documentation for this class was generated from the following file:

- Meta.h

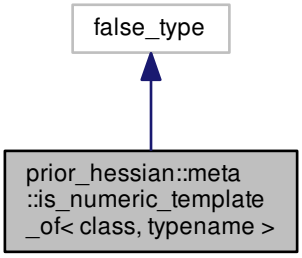
7.41 prior_hessian::meta::is_numeric_template_of< class, typename > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/Meta.h>`

Inheritance diagram for prior_hessian::meta::is_numeric_template_of< class, typename >:



Collaboration diagram for prior_hessian::meta::is_numeric_template_of< class, typename >:



7.41.1 Detailed Description

template<template< int... > class, typename>
struct prior_hessian::meta::is_numeric_template_of< class, typename >

Definition at line 86 of file Meta.h.

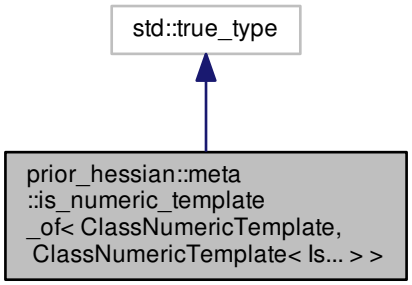
The documentation for this struct was generated from the following file:

- Meta.h

7.42 prior_hessian::meta::is_numeric_template_of< ClassNumericTemplate, ClassNumericTemplate< Is... > > Struct Template Reference

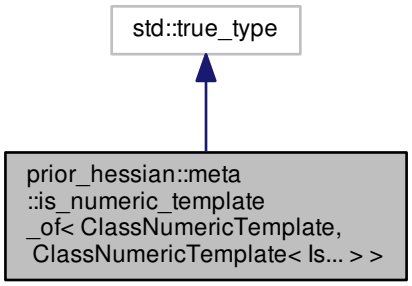
`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/Meta.h>`

Inheritance diagram for prior_hessian::meta::is_numeric_template_of< ClassNumericTemplate, ClassNumericTemplate< Is... > >:



Collaboration diagram for prior_hessian::meta::is_numeric_template_of< ClassNumericTemplate, ClassNumeric↩

Template< Is... > >:



7.42.1 Detailed Description

template<template< int... > class ClassNumericTemplate, int... Is>
struct prior_hessian::meta::is_numeric_template_of< ClassNumericTemplate, ClassNumericTemplate< Is... > >

Definition at line 89 of file Meta.h.

The documentation for this struct was generated from the following file:

- Meta.h

7.43 prior_hessian::meta::is_subclass_of_numeric_template< T, U > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ Meta.h>`

Static Public Attributes

- static constexpr bool value = decltype(is_subclass_of_numeric_template::test(std::declval<U>()))::value

7.43.1 Detailed Description

template<template< int... > class T, typename U>
class prior_hessian::meta::is_subclass_of_numeric_template< T, U >

Definition at line 92 of file Meta.h.

7.43.2 Member Data Documentation

7.43.2.1 template<template< int... > class T, typename U> constexpr bool prior_hessian::meta::is_subclass_of_↩ numeric_template< T, U >::value = decltype(is_subclass_of_numeric_template::test(std::declval<U>()))::value [static]

Definition at line 97 of file Meta.h.

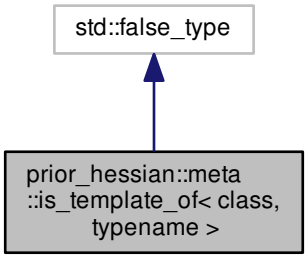
The documentation for this class was generated from the following file:

- Meta.h

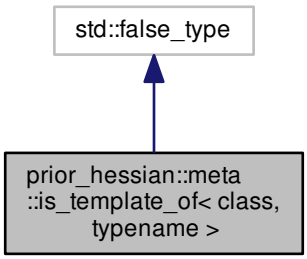
7.44 prior_hessian::meta::is_template_of< class, typename > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ Meta.h>`

Inheritance diagram for prior_hessian::meta::is_template_of< class, typename >:



Collaboration diagram for prior_hessian::meta::is_template_of< class, typename >:



7.44.1 Detailed Description

template<template< typename... > class, typename>
struct prior_hessian::meta::is_template_of< class, typename >

Definition at line 79 of file Meta.h.

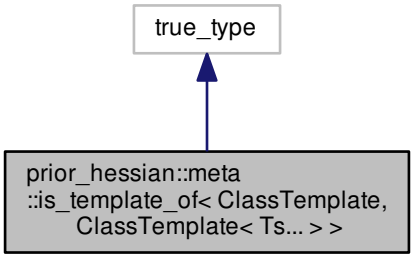
The documentation for this struct was generated from the following file:

- Meta.h

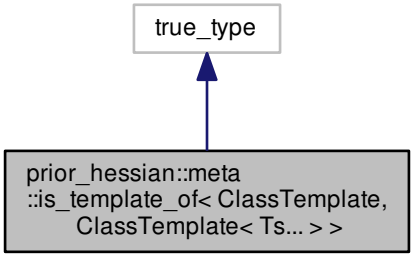
7.45 prior_hessian::meta::is_template_of< ClassTemplate, ClassTemplate< Ts... > > Struct Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/Meta.h>`

Inheritance diagram for prior_hessian::meta::is_template_of< ClassTemplate, ClassTemplate< Ts... > >:



Collaboration diagram for prior_hessian::meta::is_template_of< ClassTemplate, ClassTemplate< Ts... > >:



7.45.1 Detailed Description

template<template< typename... > class ClassTemplate, typename... Ts>
struct prior_hessian::meta::is_template_of< ClassTemplate, ClassTemplate< Ts... > >

Definition at line 82 of file Meta.h.

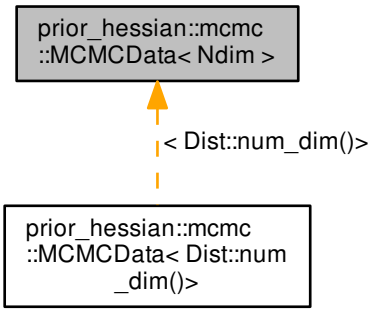
The documentation for this struct was generated from the following file:

- Meta.h

7.46 prior_hessian::mcmc::MCMCData< Ndim > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ TruncatedMultivariateDist.h>`

Inheritance diagram for prior_hessian::mcmc::MCMCData< Ndim >:



Public Types

- using NdimVecT = arma::Col< double >::fixed< Ndim >

Public Member Functions

- MCMCData ()
- MCMCData (const MCMCData< Ndim > &o)
- MCMCData< Ndim > & operator= (const MCMCData< Ndim > &o)

Public Attributes

- NdimVecT sample
- double rllh
- int nsample =0
- std::mutex mutex

7.46.1 Detailed Description

template<int Ndim>
class prior_hessian::mcmc::MCMCData< Ndim >

Definition at line 22 of file TruncatedMultivariateDist.h.

7.46.2 Member Typedef Documentation

7.46.2.1 template<int Ndim> using prior_hessian::mcmc::MCMCData< Ndim >::NdimVecT = arma::Col<double>::fixed<Ndim>

Definition at line 24 of file TruncatedMultivariateDist.h.

7.46.3 Constructor & Destructor Documentation

7.46.3.1 template<int Ndim> prior_hessian::mcmc::MCMCData< Ndim >::MCMCData ( ) [inline]

Definition at line 25 of file TruncatedMultivariateDist.h.

7.46.3.2 template<int Ndim> prior_hessian::mcmc::MCMCData< Ndim >::MCMCData (const MCMCData< Ndim > & *o*) [inline]

Definition at line 26 of file TruncatedMultivariateDist.h.

7.46.4 Member Function Documentation

7.46.4.1 template<int Ndim> MCMCData<Ndim>& prior_hessian::mcmc::MCMCData< Ndim >::operator= (const MCMCData< Ndim > & *o*) [inline]

Definition at line 37 of file TruncatedMultivariateDist.h.

7.46.5 Member Data Documentation

7.46.5.1 template<int Ndim> std::mutex prior_hessian::mcmc::MCMCData< Ndim >::mutex [mutable]

Definition at line 51 of file TruncatedMultivariateDist.h.

Referenced by prior_hessian::mcmc::MCMCData< Dist::num_dim()>::MCMCData(), and prior_hessian::mcmc::MC↩ MCData< Dist::num_dim()>::operator=().

7.46.5.2 template<int Ndim> int prior_hessian::mcmc::MCMCData< Ndim >::nsample =0

Definition at line 50 of file TruncatedMultivariateDist.h.

Referenced by prior_hessian::mcmc::MCMCData< Dist::num_dim()>::MCMCData(), and prior_hessian::mcmc::MC↩ MCData< Dist::num_dim()>::operator=().

7.46.5.3 template<int Ndim> double prior_hessian::mcmc::MCMCData< Ndim >::rllh

Definition at line 49 of file TruncatedMultivariateDist.h.

Referenced by prior_hessian::mcmc::MCMCData< Dist::num_dim()>::MCMCData(), prior_hessian::mcmc::MCMC↩ Data< Dist::num_dim()>::operator=(), and prior_hessian::TruncatedMultivariateDist< Dist >::sample().

7.46.5.4 template<int Ndim> NdimVecT prior_hessian::mcmc::MCMCData< Ndim >::sample

Definition at line 48 of file TruncatedMultivariateDist.h.

Referenced by prior_hessian::mcmc::MCMCData< Dist::num_dim()>::MCMCData(), prior_hessian::Truncated↩ MultivariateDist< Dist >::mean(), prior_hessian::mcmc::MCMCData< Dist::num_dim()>::operator=(), and prior_↩ hessian::TruncatedMultivariateDist< Dist >::sample().

The documentation for this class was generated from the following file:

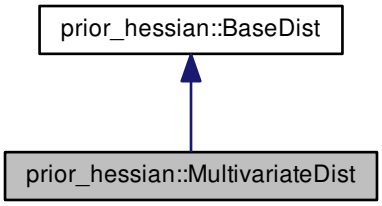
- TruncatedMultivariateDist.h

7.47 prior_hessian::MultivariateDist Class Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ MultivariateDist.h>`

Inheritance diagram for prior_hessian::MultivariateDist:

Collaboration diagram for prior_hessian::MultivariateDist:



Public Member Functions

- MultivariateDist ()

Static Protected Member Functions

- template<class Vec >
  static void check_bounds (const Vec &lbound, const Vec &ubound)

7.47.1 Detailed Description

Definition at line 15 of file MultivariateDist.h.

7.47.2 Constructor & Destructor Documentation

7.47.2.1 prior_hessian::MultivariateDist::MultivariateDist ( ) [inline]

Definition at line 18 of file MultivariateDist.h.

7.47.3 Member Function Documentation

7.47.3.1 template<class Vec > static void prior_hessian::MultivariateDist::check_bounds (const Vec & *lbound,* const Vec & *ubound*) [inline], [static], [protected]

Definition at line 57 of file MultivariateDist.h.

The documentation for this class was generated from the following file:

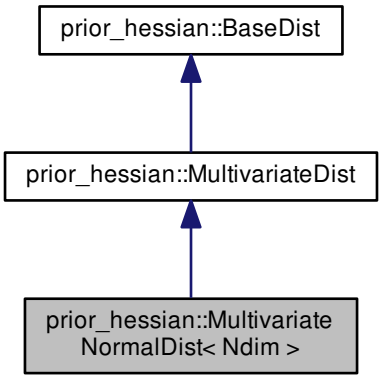
- MultivariateDist.h

7.48 prior_hessian::MultivariateNormalDist< Ndim > Class Template Reference

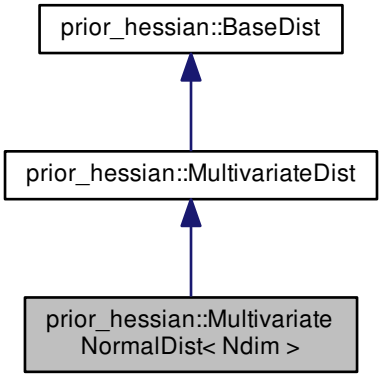
Multivariate Normal distribution.

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/MultivariateNormalDist.h>`

Inheritance diagram for prior_hessian::MultivariateNormalDist< Ndim >:



Collaboration diagram for prior_hessian::MultivariateNormalDist< Ndim >:



Public Types

- using NdimVecT = arma::Col< double >::fixed< Ndim >
- using NdimMatT = arma::Mat< double >::fixed< Ndim, Ndim >
- using NparamsVecT = arma::Col< double >::fixed< _num_params >

Public Member Functions

- MultivariateNormalDist ()
- template<class Vec , class Mat >
  MultivariateNormalDist (Vec &&mu, Mat &&sigma)
- const NdimVecT & mu () const
- const NdimMatT & sigma () const
- const NdimMatT & sigma_inv () const
- template<class Vec >
  void set_mu (Vec &&val)
- template<class Mat >
  void set_sigma (Mat &&val)
- bool operator== (const MultivariateNormalDist< Ndim > &o) const
- bool operator!= (const MultivariateNormalDist< Ndim > &o) const
- double get_param (IdxT idx) const
- NparamsVecT params () const
- template<class Vec >
  void set_params (const Vec &p)
- template<class Vec , class Mat >
  void set_params (Vec &&mu, Mat &&sigma)
- NdimVecT mean () const
- NdimVecT mode () const
- template<class Vec >
  double cdf (Vec x) const
- template<class Vec >
  double pdf (const Vec &x) const
- template<class Vec >
  double llh (const Vec &x) const
- template<class Vec >
  double rllh (const Vec &x) const
- template<class Vec >
  NdimVecT grad (const Vec &x) const
- template<class Vec >
  NdimVecT grad2 (const Vec &x) const
- template<class Vec >
  NdimMatT hess (const Vec &x) const
- template<class Vec , class Vec2 >
  void grad_grad2_accumulate (const Vec &x, Vec2 &g, Vec2 &g2) const
- template<class Vec , class Vec2 , class Mat >
  void grad_hess_accumulate (const Vec &x, Vec2 &g, Mat &hess) const
- template<class RngT >
  NdimVecT sample (RngT &rng) const
- template<class IterT >
  void append_params (IterT &params) const
- template<class IterT >
  void set_params_iter (IterT &params)
- template<>
  double cdf (Vec x) const

Static Public Member Functions

- static constexpr IdxT num_params ()
- static constexpr IdxT num_dim ()
- static const NdimVecT & lbound ()
- static const NdimVecT & ubound ()
- template<class Vec >
  static bool in_bounds (const Vec &u)
- static const StringVecT & param_names ()
- static const NparamsVecT & param_lbound ()
- static const NparamsVecT & param_ubound ()
- template<class Vec >
  static bool check_mu (const Vec &mu)
- template<class Mat >
  static bool check_sigma (const Mat &sigma)
- template<class Vec , class Mat >
  static bool check_params (const Vec &mu, const Mat &sigma)
- template<class Vec >
  static bool check_params (const Vec &params)
- template<class IterT >
  static bool check_params_iter (IterT &params)

Static Protected Member Functions

- template<class Vec >
  static void check_bounds (const Vec &lbound, const Vec &ubound)

7.48.1 Detailed Description

template<IdxT Ndim>
class prior_hessian::MultivariateNormalDist< Ndim >

Multivariate Normal distribution.

Definition at line 18 of file MultivariateNormalDist.h.

7.48.2 Member Typedef Documentation

7.48.2.1 template<IdxT Ndim> using prior_hessian::MultivariateNormalDist< Ndim >::NdimMatT = arma::Mat<double>::fixed<Ndim,Ndim>

Definition at line 24 of file MultivariateNormalDist.h.

7.48.2.2 template<IdxT Ndim> using prior_hessian::MultivariateNormalDist< Ndim >::NdimVecT = arma::Col<double>::fixed<Ndim>

Definition at line 23 of file MultivariateNormalDist.h.

7.48.2.3 template<IdxT Ndim> using prior_hessian::MultivariateNormalDist< Ndim >::NparamsVecT = arma::Col<double>::fixed<_num_params>

Definition at line 25 of file MultivariateNormalDist.h.

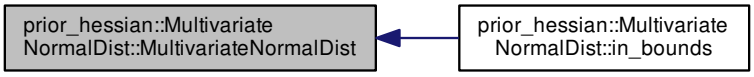
7.48.3 Constructor & Destructor Documentation

7.48.3.1 template<IdxT Ndim> prior_hessian::MultivariateNormalDist< Ndim >::MultivariateNormalDist ( )

Definition at line 172 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds().

Here is the caller graph for this function:

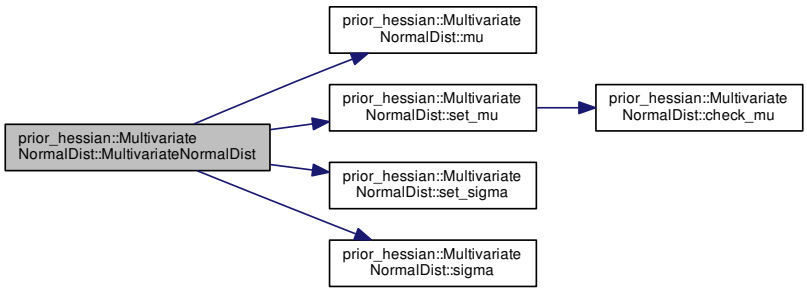


7.48.3.2 template<IdxT Ndim> template<class Vec , class Mat > prior_hessian::MultivariateNormalDist< Ndim >::MultivariateNormalDist (Vec && *mu,* Mat && *sigma*)

Definition at line 185 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::mu(), prior_hessian::MultivariateNormalDist< Ndim >::set_mu(), prior_hessian::MultivariateNormalDist< Ndim >::set_sigma(), and prior_hessian::MultivariateNormalDist< Ndim >::sigma().

Here is the call graph for this function:

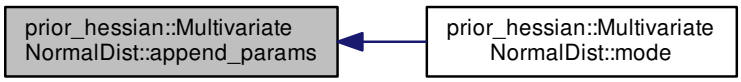


7.48.4 Member Function Documentation

7.48.4.1 template<IdxT Ndim> template<class IterT > void prior_hessian::MultivariateNormalDist< Ndim >::append_params (IterT & *params*) const

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the caller graph for this function:



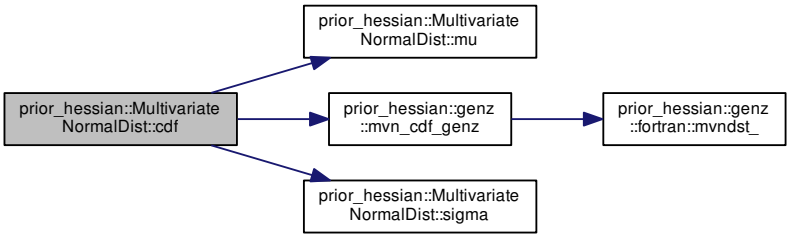
7.48.4.2 template<IdxT Ndim> template<class Vec > double prior_hessian::MultivariateNormalDist< Ndim >::cdf (Vec *x*) const

Definition at line 480 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::mu(), prior_hessian::genz::mvn_cdf_genz(), and prior_↩ hessian::MultivariateNormalDist< Ndim >::sigma().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:



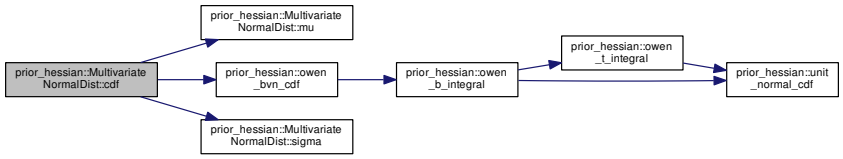
Here is the caller graph for this function:

7.48.4.3 template<> double prior_hessian::MultivariateNormalDist< 2 >::cdf (Vec *x*) const

Definition at line 489 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::mu(), prior_hessian::owen_bvn_cdf(), and prior_hessian::MultivariateNormalDist< Ndim >::sigma().

Here is the call graph for this function:



7.48.4.4 template<class Vec > static void prior_hessian::MultivariateDist::check_bounds (const Vec & *lbound,* const Vec & *ubound*) [inline],[static],[protected],[inherited]

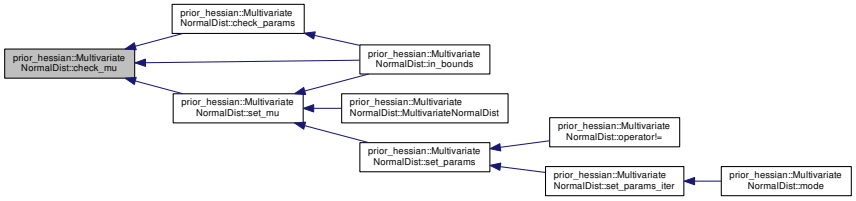
Definition at line 57 of file MultivariateDist.h.

7.48.4.5 template<IdxT Ndim> template<class Vec > bool prior_hessian::MultivariateNormalDist< Ndim >::check_mu (const Vec & *mu*) [static]

Definition at line 194 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::check_params(), prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), and prior_hessian::MultivariateNormalDist< Ndim >::set_mu().

Here is the caller graph for this function:



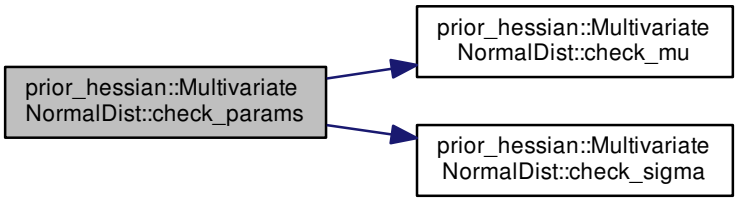
7.48.4.6 template<IdxT Ndim> template<class Vec , class Mat > bool prior_hessian::MultivariateNormalDist< Ndim >::check_params (const Vec & *mu,* const Mat & *sigma*) [static]

Definition at line 211 of file MultivariateNormalDist.h.

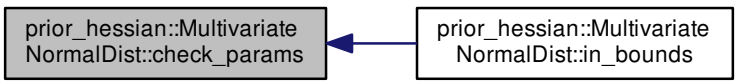
References prior_hessian::MultivariateNormalDist< Ndim >::check_mu(), and prior_hessian::MultivariateNormalDist< Ndim >::check_sigma().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds().

Here is the call graph for this function:



Here is the caller graph for this function:

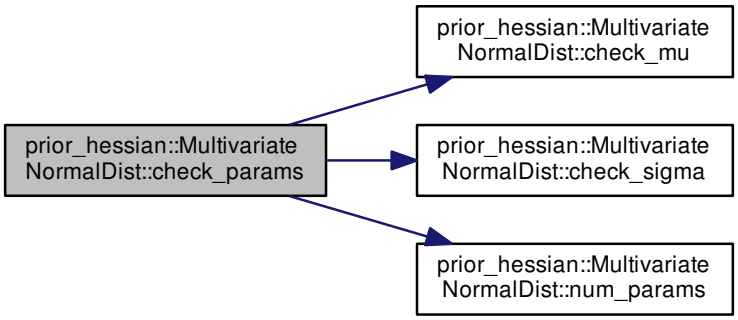


7.48.4.7 template<IdxT Ndim> template<class Vec > bool prior_hessian::MultivariateNormalDist< Ndim >::check_params (const Vec & *params*) [static]

Definition at line 218 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::check_mu(), prior_hessian::MultivariateNormalDist< Ndim >::check_sigma(), and prior_hessian::MultivariateNormalDist< Ndim >::num_params().

Here is the call graph for this function:



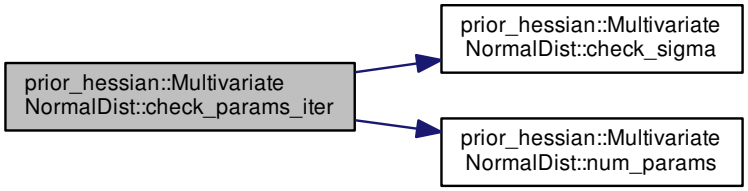
7.48.4.8 template<IdxT Ndim> template<class IterT > bool prior_hessian::MultivariateNormalDist< Ndim >::check_params_iter (IterT & *params*) [static]

Definition at line 226 of file MultivariateNormalDist.h.

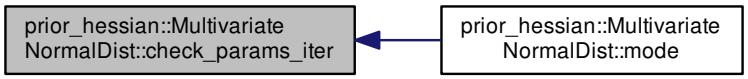
References prior_hessian::MultivariateNormalDist< Ndim >::check_sigma(), and prior_hessian::MultivariateNormal↩ Dist< Ndim >::num_params().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:



Here is the caller graph for this function:

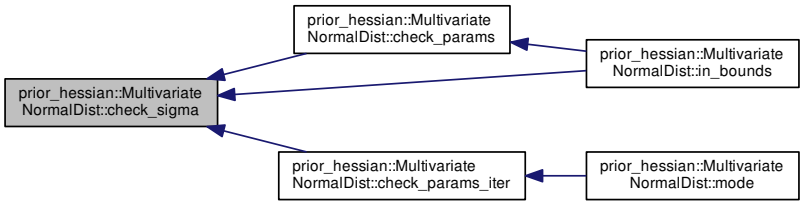


7.48.4.9 template<IdxT Ndim> template<class Mat > bool prior_hessian::MultivariateNormalDist< Ndim >::check_sigma (const Mat & *sigma*) [static]

Definition at line 201 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::check_params(), prior_hessian::MultivariateNormal↩ Dist< Ndim >::check_params_iter(), and prior_hessian::MultivariateNormalDist< Ndim >::in_bounds().

Here is the caller graph for this function:



7.48.4.10 template<IdxT Ndim> double prior_hessian::MultivariateNormalDist< Ndim >::get_param (IdxT *idx*) const

Definition at line 440 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::operator!=().

Here is the caller graph for this function:

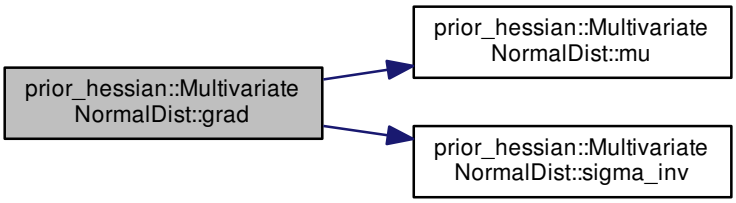
7.48.4.11 template<IdxT Ndim> template<class Vec > MultivariateNormalDist< Ndim >::NdimVecT prior_hessian::MultivariateNormalDist< Ndim >::grad (const Vec & *x*) const

Definition at line 522 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::mu(), and prior_hessian::MultivariateNormalDist< Ndim >::sigma_inv().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:



Here is the caller graph for this function:

7.48.4.12 template<IdxT Ndim> template<class Vec > MultivariateNormalDist< Ndim >::NdimVecT prior_hessian::MultivariateNormalDist< Ndim >::grad2 (const Vec & *x*) const

Definition at line 530 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::sigma_inv().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:

Here is the caller graph for this function:

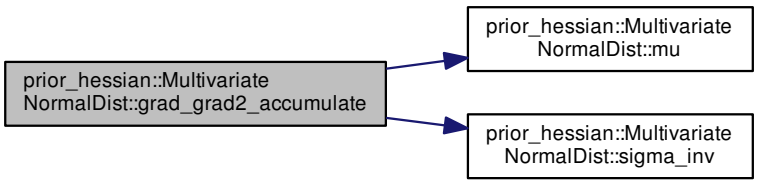
7.48.4.13 template<IdxT Ndim> template<class Vec , class Vec2 > void prior_hessian::MultivariateNormalDist< Ndim >::grad_grad2_accumulate (const Vec & *x*, Vec2 & *g*, Vec2 & *g2*) const

Definition at line 545 of file MultivariateNormalDist.h.

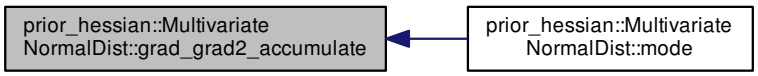
References prior_hessian::MultivariateNormalDist< Ndim >::mu(), and prior_hessian::MultivariateNormalDist< Ndim >::sigma_inv().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:



Here is the caller graph for this function:



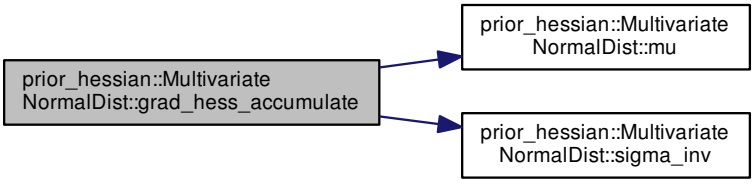
7.48.4.14 template<IdxT Ndim> template<class Vec , class Vec2 , class Mat > void prior_hessian::MultivariateNormalDist< Ndim >::grad_hess_accumulate (const Vec & *x*, Vec2 & *g*, Mat & *hess*) const

Definition at line 553 of file MultivariateNormalDist.h.

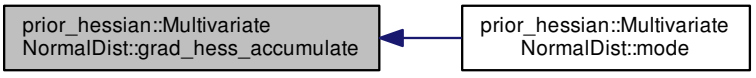
References prior_hessian::MultivariateNormalDist< Ndim >::mu(), and prior_hessian::MultivariateNormalDist< Ndim >::sigma_inv().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:



Here is the caller graph for this function:



7.48.4.15 template<IdxT Ndim> template<class Vec > MultivariateNormalDist< Ndim >::NdimMatT prior_hessian::MultivariateNormalDist< Ndim >::hess (const Vec & *x*) const

Definition at line 538 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::sigma_inv().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:

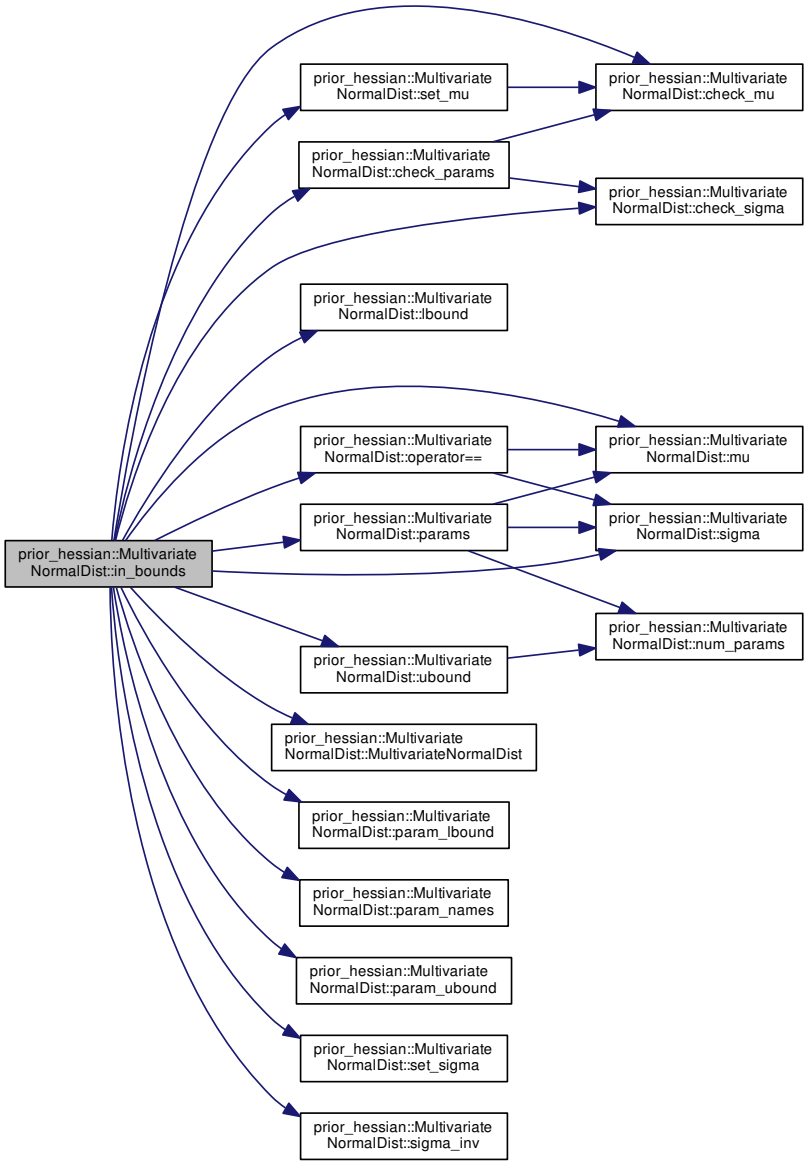
Here is the caller graph for this function:

7.48.4.16 template<IdxT Ndim> template<class Vec > static bool prior_hessian::MultivariateNormalDist< Ndim >::in_bounds (const Vec & *u*) [inline], [static]

Definition at line 32 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::check_mu(), prior_hessian::MultivariateNormalDist< Ndim >::check_params(), prior_hessian::MultivariateNormalDist< Ndim >::check_sigma(), prior_hessian::MultivariateNormalDist< Ndim >::lbound(), prior_hessian::MultivariateNormalDist< Ndim >::mu(), prior_hessian::MultivariateNormalDist< Ndim >::MultivariateNormalDist(), prior_hessian::MultivariateNormalDist< Ndim >::operator==(), prior_hessian::MultivariateNormalDist< Ndim >::param_lbound(), prior_hessian::MultivariateNormalDist< Ndim >::param_names(), prior_hessian::MultivariateNormalDist< Ndim >::param_ubound(), prior_hessian::MultivariateNormalDist< Ndim >::params(), prior_hessian::MultivariateNormalDist< Ndim >::set_mu(), prior_hessian::MultivariateNormalDist< Ndim >::set_sigma(), prior_hessian::MultivariateNormalDist< Ndim >::sigma(), prior_hessian::MultivariateNormalDist< Ndim >::sigma_inv(), and prior_hessian::MultivariateNormalDist< Ndim >::ubound().

Here is the call graph for this function:

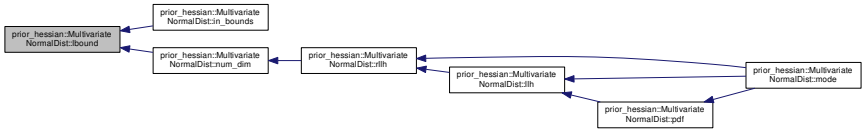


7.48.4.17 template<IdxT Ndim> const MultivariateNormalDist< Ndim >::NdimVecT & prior_hessian::MultivariateNormalDist< Ndim >::lbound ( ) [static]

Definition at line 296 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), and prior_hessian::MultivariateNormal↩Dist< Ndim >::num_dim().

Here is the caller graph for this function:



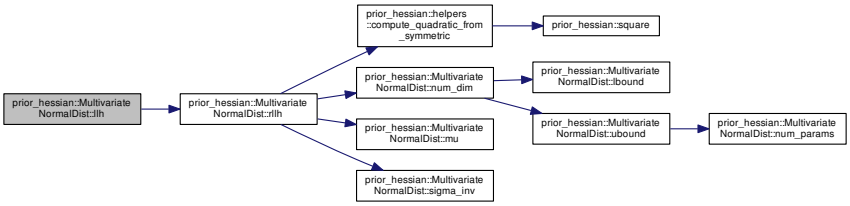
7.48.4.18 template<IdxT Ndim> template<class Vec > double prior_hessian::MultivariateNormalDist< Ndim >::llh (const Vec & *x*) const

Definition at line 506 of file MultivariateNormalDist.h.

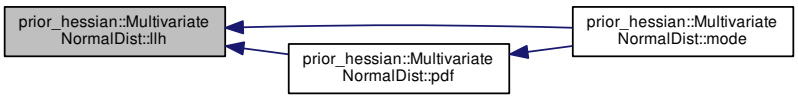
References prior_hessian::MultivariateNormalDist< Ndim >::rllh().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode(), and prior_hessian::MultivariateNormalDist< Ndim >::pdf().

Here is the call graph for this function:



Here is the caller graph for this function:



7.48.4.19 template<IdxT Ndim> NdimVecT prior_hessian::MultivariateNormalDist< Ndim >::mean ( ) const [inline]

Definition at line 71 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::mu().

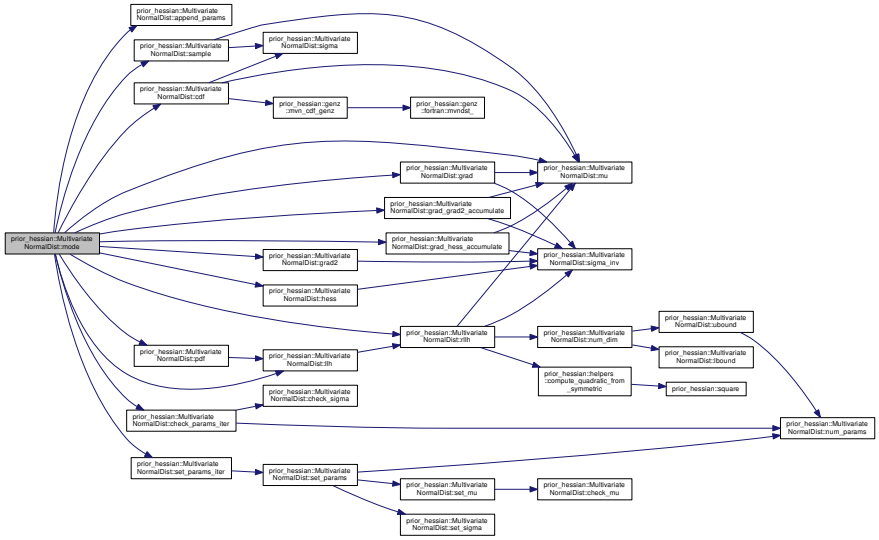
Here is the call graph for this function:

7.48.4.20 template<IdxT Ndim> NdimVecT prior_hessian::MultivariateNormalDist< Ndim >::mode ( ) const [inline]

Definition at line 72 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::append_params(), prior_hessian::MultivariateNormal↩ Dist< Ndim >::cdf(), prior_hessian::MultivariateNormalDist< Ndim >::check_params_iter(), prior_hessian::↩ MultivariateNormalDist< Ndim >::grad(), prior_hessian::MultivariateNormalDist< Ndim >::grad2(), prior_hessian↩ ::MultivariateNormalDist< Ndim >::grad_grad2_accumulate(), prior_hessian::MultivariateNormalDist< Ndim >::grad↩ _hess_accumulate(), prior_hessian::MultivariateNormalDist< Ndim >::hess(), prior_hessian::MultivariateNormalDist< Ndim >::llh(), prior_hessian::MultivariateNormalDist< Ndim >::mu(), prior_hessian::MultivariateNormalDist< Ndim >::pdf(), prior_hessian::MultivariateNormalDist< Ndim >::rllh(), prior_hessian::MultivariateNormalDist< Ndim >↩ ::sample(), and prior_hessian::MultivariateNormalDist< Ndim >::set_params_iter().

Here is the call graph for this function:

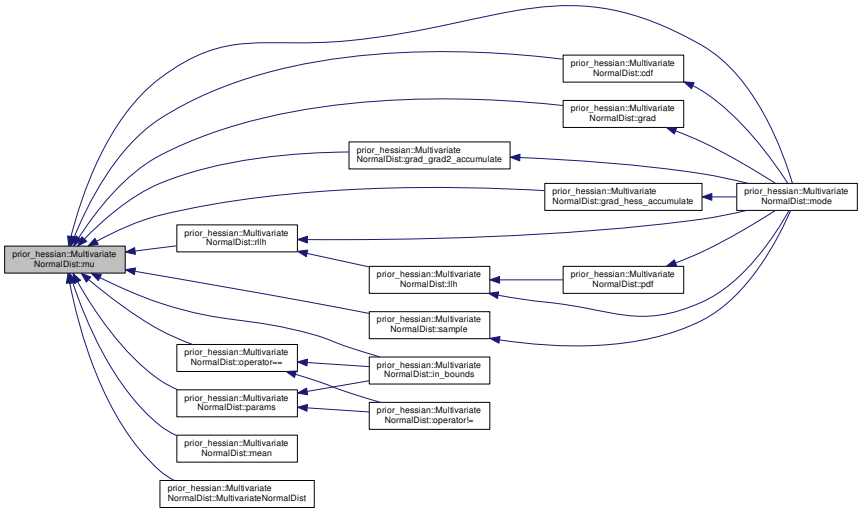


7.48.4.21 template<IdxT Ndim> const MultivariateNormalDist< Ndim >::NdimVecT & prior_hessian::MultivariateNormalDist< Ndim >::mu ( ) const

Definition at line 374 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::cdf(), prior_hessian::MultivariateNormalDist< Ndim >::grad(), prior_hessian::MultivariateNormalDist< Ndim >::grad_grad2_accumulate(), prior_hessian::MultivariateNormalDist< Ndim >::grad_hess_accumulate(), prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), prior_hessian::MultivariateNormalDist< Ndim >::mean(), prior_hessian::MultivariateNormalDist< Ndim >::mode(), prior_hessian::MultivariateNormalDist< Ndim >::MultivariateNormalDist(), prior_hessian::MultivariateNormalDist< Ndim >::operator==(), prior_hessian::MultivariateNormalDist< Ndim >::params(), prior_hessian::MultivariateNormalDist< Ndim >::rllh(), and prior_hessian::MultivariateNormalDist< Ndim >::sample().

Here is the caller graph for this function:



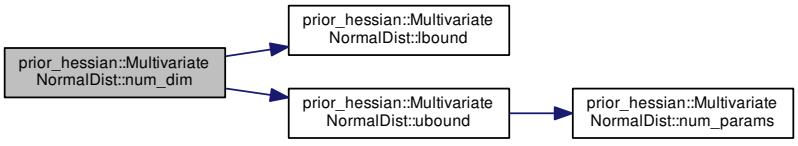
7.48.4.22 template<IdxT Ndim> static constexpr IdxT prior_hessian::MultivariateNormalDist< Ndim >::num_dim ( ) [inline], [static]

Definition at line 28 of file MultivariateNormalDist.h.

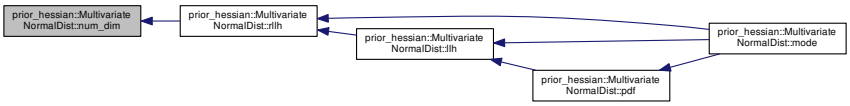
References prior_hessian::MultivariateNormalDist< Ndim >::lbound(), and prior_hessian::MultivariateNormalDist< Ndim >::ubound().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::rllh().

Here is the call graph for this function:



Here is the caller graph for this function:

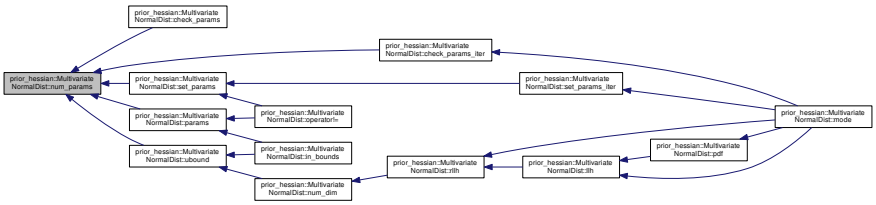


7.48.4.23 template<IdxT Ndim> static constexpr IdxT prior_hessian::MultivariateNormalDist< Ndim >::num_params ( ) [inline], [static]

Definition at line 27 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::check_params(), prior_hessian::MultivariateNormal↩ Dist< Ndim >::check_params_iter(), prior_hessian::MultivariateNormalDist< Ndim >::params(), prior_hessian::↩ MultivariateNormalDist< Ndim >::set_params(), and prior_hessian::MultivariateNormalDist< Ndim >::ubound().

Here is the caller graph for this function:

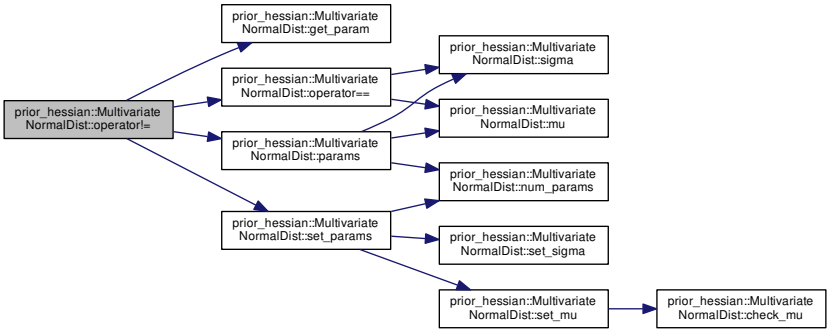


7.48.4.24 template<IdxT Ndim> bool prior_hessian::MultivariateNormalDist< Ndim >::operator!= (const MultivariateNormalDist< Ndim > & *o*) const [inline]

Definition at line 59 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::get_param(), prior_hessian::MultivariateNormalDist< Ndim >::operator==(), prior_hessian::MultivariateNormalDist< Ndim >::params(), and prior_hessian::Multivariate↩ NormalDist< Ndim >::set_params().

Here is the call graph for this function:



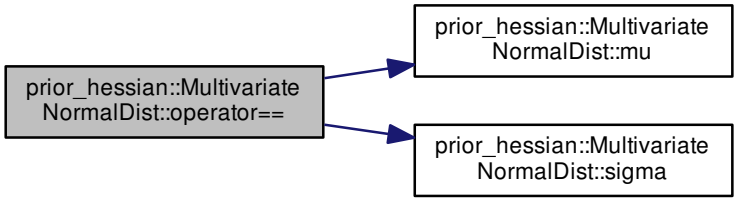
7.48.4.25 template<IdxT Ndim> bool prior_hessian::MultivariateNormalDist< Ndim >::operator== (const MultivariateNormalDist< Ndim > & *o*) const

Definition at line 424 of file MultivariateNormalDist.h.

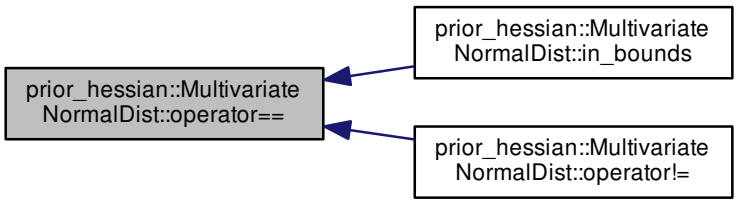
References prior_hessian::MultivariateNormalDist< Ndim >::mu(), and prior_hessian::MultivariateNormalDist< Ndim >::sigma().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), and prior_hessian::MultivariateNormal↩ Dist< Ndim >::operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

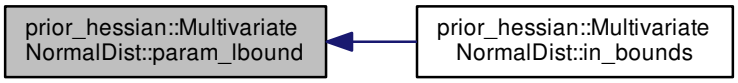


7.48.4.26 template<IdxT Ndim> const MultivariateNormalDist< Ndim >::NparamsVecT & prior_hessian::MultivariateNormalDist< Ndim >::param_lbound ( ) [static]

Definition at line 278 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds().

Here is the caller graph for this function:



7.48.4.27 template<IdxT Ndim> const StringVecT & prior_hessian::MultivariateNormalDist< Ndim >::param_names ( ) [static]

Definition at line 269 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds().

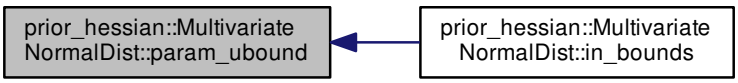
Here is the caller graph for this function:

7.48.4.28 template<IdxT Ndim> const MultivariateNormalDist< Ndim >::NparamsVecT & prior_hessian::MultivariateNormalDist< Ndim >::param_ubound ( ) [static]

Definition at line 287 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds().

Here is the caller graph for this function:



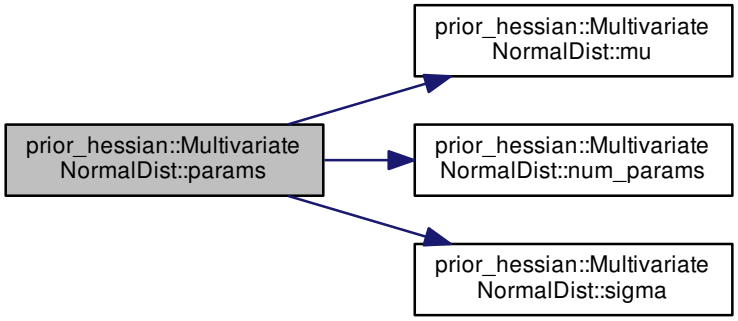
7.48.4.29 template<IdxT Ndim> MultivariateNormalDist< Ndim >::NparamsVecT prior_hessian::MultivariateNormalDist< Ndim >::params ( ) const

Definition at line 431 of file MultivariateNormalDist.h.

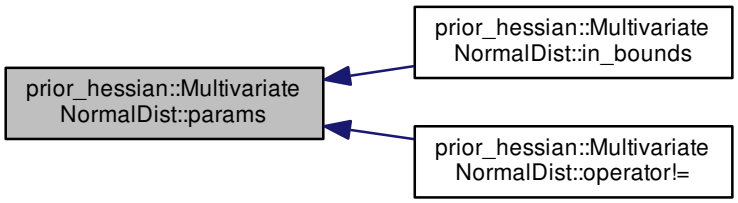
References prior_hessian::MultivariateNormalDist< Ndim >::mu(), prior_hessian::MultivariateNormalDist< Ndim >::num_params(), and prior_hessian::MultivariateNormalDist< Ndim >::sigma().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), and prior_hessian::MultivariateNormalDist< Ndim >::operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:



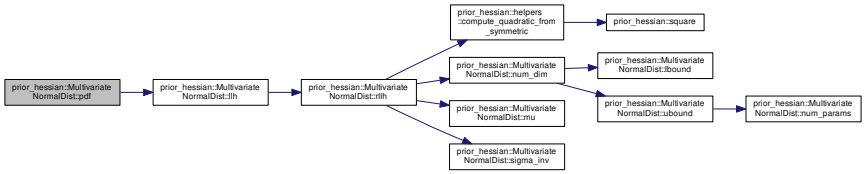
7.48.4.30 template<IdxT Ndim> template<class Vec > double prior_hessian::MultivariateNormalDist< Ndim >::pdf (const Vec & *x*) const

Definition at line 496 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::llh().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:



Here is the caller graph for this function:

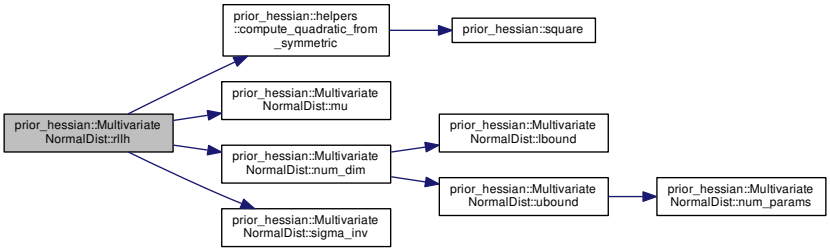
7.48.4.31 template<IdxT Ndim> template<class Vec > double prior_hessian::MultivariateNormalDist< Ndim >::rllh (const Vec & *x*) const

Definition at line 514 of file MultivariateNormalDist.h.

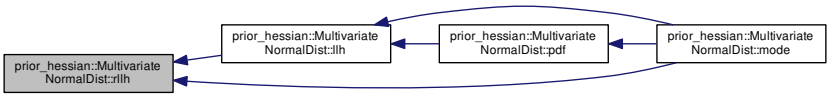
References prior_hessian::helpers::compute_quadratic_from_symmetric(), prior_hessian::MultivariateNormalDist< Ndim >::mu(), prior_hessian::MultivariateNormalDist< Ndim >::num_dim(), and prior_hessian::MultivariateNormal↩ Dist< Ndim >::sigma_inv().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::llh(), and prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:



Here is the caller graph for this function:



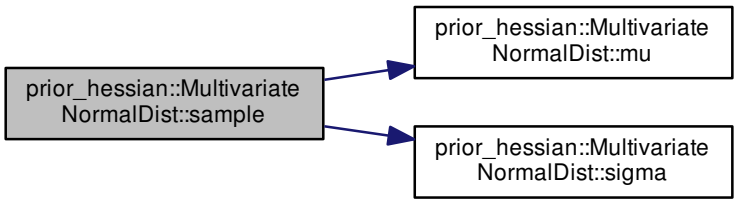
7.48.4.32 template<IdxT Ndim> template<class RngT > MultivariateNormalDist< Ndim >::NdimVecT prior_hessian::MultivariateNormalDist< Ndim >::sample (RngT & *rng*) const

Definition at line 562 of file MultivariateNormalDist.h.

References prior_hessian::constants::log2pi, prior_hessian::MultivariateNormalDist< Ndim >::mu(), and prior_hessian::MultivariateNormalDist< Ndim >::sigma().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:



Here is the caller graph for this function:

7.48.4.33 template<IdxT Ndim> template<class Vec > void prior_hessian::MultivariateNormalDist< Ndim >::set_mu (Vec && *val*)

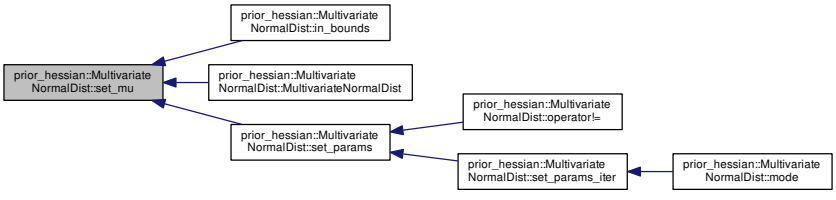
Definition at line 394 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::check_mu().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), prior_hessian::MultivariateNormalDist< Ndim >::MultivariateNormalDist(), and prior_hessian::MultivariateNormalDist< Ndim >::set_params().

Here is the call graph for this function:

Here is the caller graph for this function:



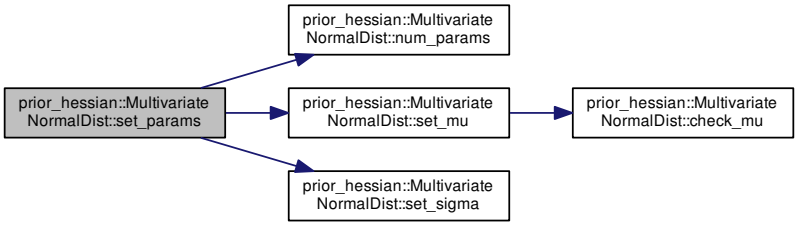
7.48.4.34 template<IdxT Ndim> template<class Vec > void prior_hessian::MultivariateNormalDist< Ndim >::set_params (const Vec & *p*)

Definition at line 452 of file MultivariateNormalDist.h.

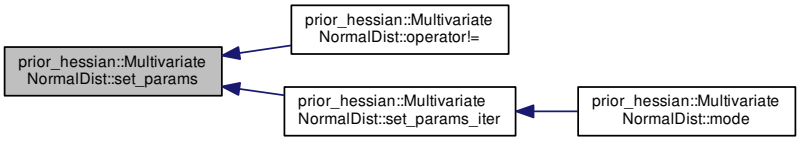
References prior_hessian::MultivariateNormalDist< Ndim >::num_params(), prior_hessian::MultivariateNormalDist< Ndim >::set_mu(), and prior_hessian::MultivariateNormalDist< Ndim >::set_sigma().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::operator!=(), and prior_hessian::MultivariateNormal↩ Dist< Ndim >::set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:

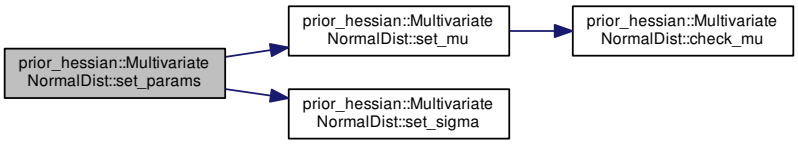


7.48.4.35 template<IdxT Ndim> template<class Vec , class Mat > void prior_hessian::MultivariateNormalDist< Ndim >::set_params (Vec && *mu,* Mat && *sigma*)

Definition at line 460 of file MultivariateNormalDist.h.

References prior_hessian::MultivariateNormalDist< Ndim >::set_mu(), and prior_hessian::MultivariateNormalDist< Ndim >::set_sigma().

Here is the call graph for this function:



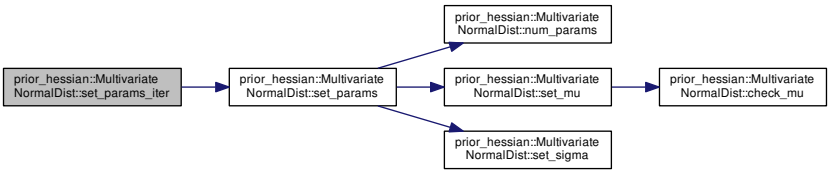
7.48.4.36 template<IdxT Ndim> template<class IterT > void prior_hessian::MultivariateNormalDist< Ndim >::set_params_iter (IterT & *params*)

Definition at line 468 of file MultivariateNormalDist.h.

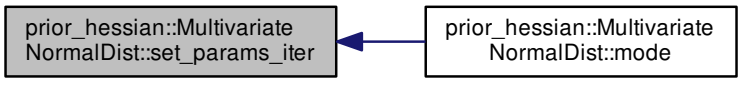
References prior_hessian::MultivariateNormalDist< Ndim >::set_params().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::mode().

Here is the call graph for this function:



Here is the caller graph for this function:

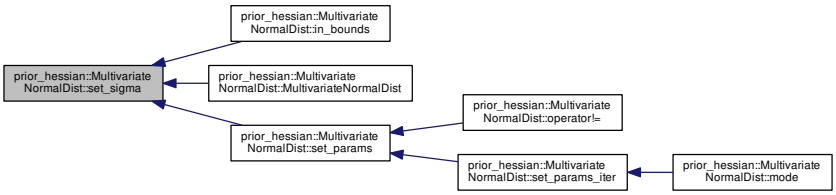


7.48.4.37 template<IdxT Ndim> template<class Mat > void prior_hessian::MultivariateNormalDist< Ndim >::set_sigma (Mat && *val*)

Definition at line 399 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), prior_hessian::MultivariateNormalDist< Ndim >::MultivariateNormalDist(), and prior_hessian::MultivariateNormalDist< Ndim >::set_params().

Here is the caller graph for this function:

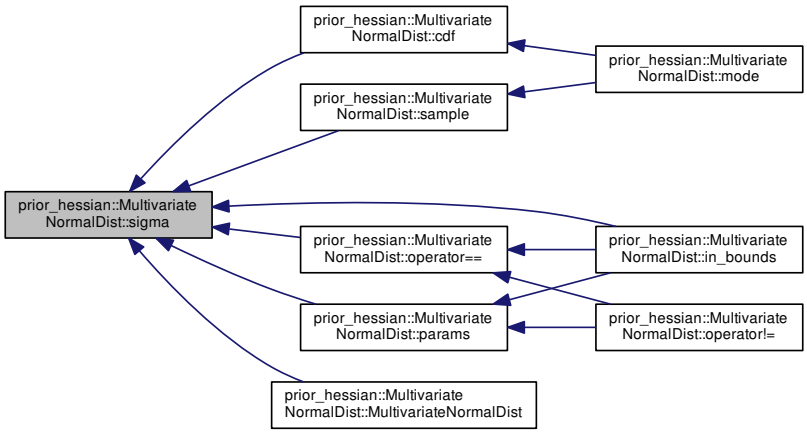


7.48.4.38 template<IdxT Ndim> const MultivariateNormalDist< Ndim >::NdimMatT & prior_hessian::MultivariateNormalDist< Ndim >::sigma ( ) const

Definition at line 379 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::cdf(), prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), prior_hessian::MultivariateNormalDist< Ndim >::MultivariateNormalDist(), prior_hessian↩ ::MultivariateNormalDist< Ndim >::operator==(), prior_hessian::MultivariateNormalDist< Ndim >::params(), and prior_hessian::MultivariateNormalDist< Ndim >::sample().

Here is the caller graph for this function:

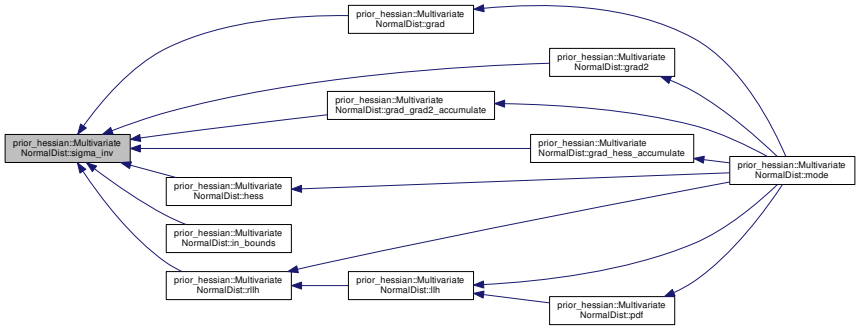


7.48.4.39 template<IdxT Ndim> const MultivariateNormalDist< Ndim >::NdimMatT & prior_hessian::MultivariateNormalDist< Ndim >::sigma_inv ( ) const

Definition at line 384 of file MultivariateNormalDist.h.

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::grad(), prior_hessian::MultivariateNormalDist< Ndim >::grad2(), prior_hessian::MultivariateNormalDist< Ndim >::grad_grad2_accumulate(), prior_hessian::MultivariateNormalDist< Ndim >::grad_hess_accumulate(), prior_hessian::MultivariateNormalDist< Ndim >::hess(), prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), and prior_hessian::MultivariateNormalDist< Ndim >::rllh().

Here is the caller graph for this function:



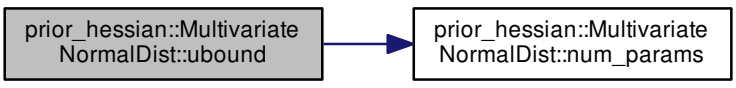
7.48.4.40 template<IdxT Ndim> const MultivariateNormalDist< Ndim >::NdimVecT & prior_hessian::MultivariateNormalDist< Ndim >::ubound ( ) [static]

Definition at line 305 of file MultivariateNormalDist.h.

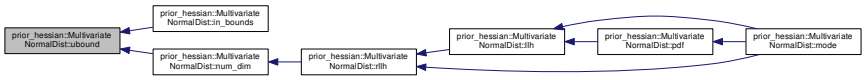
References prior_hessian::MultivariateNormalDist< Ndim >::num_params().

Referenced by prior_hessian::MultivariateNormalDist< Ndim >::in_bounds(), and prior_hessian::MultivariateNormalDist< Ndim >::num_dim().

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following file:

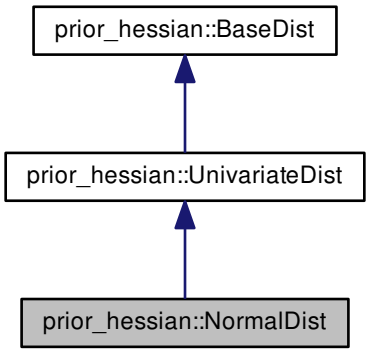
- MultivariateNormalDist.h

7.49 prior_hessian::NormalDist Class Reference

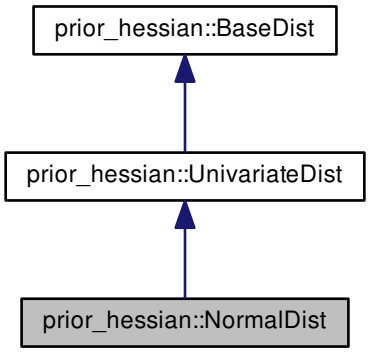
Normal distribution with truncation.

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ NormalDist.h>`

Inheritance diagram for prior_hessian::NormalDist:



Collaboration diagram for prior_hessian::NormalDist:



Public Types

- using NparamsVecT = arma::Col< double >::fixed< _num_params >

Public Member Functions

- NormalDist (double mu, double sigma)
- NormalDist ()
- template<class Vec , meta::ConstructableIfNotSelfT< Vec, NormalDist > = true>
  NormalDist (const Vec &params)
- double mu () const
- double sigma () const
- void set_mu (double val)
- void set_sigma (double val)
- bool operator== (const NormalDist &o) const
- bool operator!= (const NormalDist &o) const
- double get_param (int idx) const
- void set_param (int idx, double val)
- NparamsVecT params () const
- void set_params (double mu, double sigma)
- template<class Vec >
  void set_params (const Vec &p)
- double mean () const
- double median () const
- double cdf (double x) const
- double icdf (double u) const
- double pdf (double x) const
- double llh (double x) const
- double rllh (double x) const

- double grad (double x) const
- double grad2 (double x) const
- void grad_grad2_accumulate (double x, double &g, double &g2) const
- template<class RngT >
  double sample (RngT &rng) const
- template<class IterT >
  void append_params_iter (IterT &params) const
- template<class IterT >
  void set_params_iter (IterT &params)

Static Public Member Functions

- static constexpr IdxT num_params ()
- static constexpr double lbound ()
- static constexpr double ubound ()
- static bool in_bounds (double u)
- static const StringVecT & param_names ()
- static const NparamsVecT & param_lbound ()
- static const NparamsVecT & param_ubound ()
- static bool check_params (double mu, double sigma)
- template<class Vec >
  static bool check_params (const Vec &p)
- template<class IterT >
  static bool check_params_iter (IterT &params)
- static constexpr IdxT num_dim ()

Static Protected Member Functions

- static void check_bounds (double lbound, double ubound)

7.49.1 Detailed Description

Normal distribution with truncation.

Definition at line 21 of file NormalDist.h.

7.49.2 Member Typedef Documentation

7.49.2.1 using prior_hessian::NormalDist::NparamsVecT = arma::Col<double>::fixed<_num_params>

Definition at line 25 of file NormalDist.h.

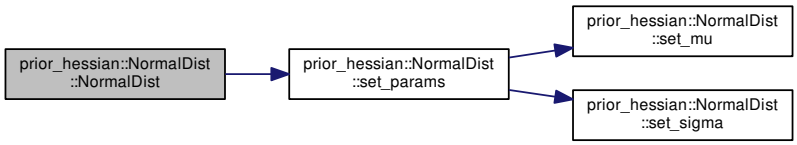
7.49.3 Constructor & Destructor Documentation

7.49.3.1 prior_hessian::NormalDist::NormalDist (double *mu,* double *sigma*)

Definition at line 27 of file NormalDist.cpp.

References set_params().

Here is the call graph for this function:

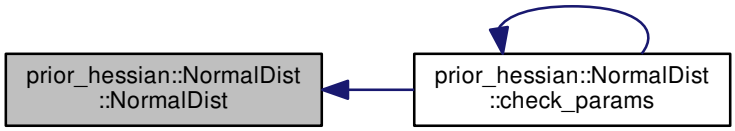


7.49.3.2 prior_hessian::NormalDist::NormalDist ( ) [inline]

Definition at line 43 of file NormalDist.h.

Referenced by check_params().

Here is the caller graph for this function:



7.49.3.3 template<class Vec , meta::ConstructableIfNotSelfT< Vec, NormalDist > = true> prior_hessian::NormalDist::NormalDist (const Vec & *params*) [inline], [explicit]

Definition at line 45 of file NormalDist.h.

7.49.4 Member Function Documentation

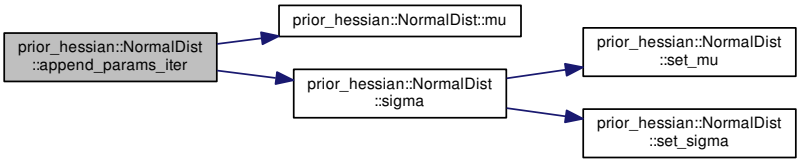
7.49.4.1 template<class IterT > void prior_hessian::NormalDist::append_params_iter (IterT & *params*) const

Definition at line 205 of file NormalDist.h.

References mu(), and sigma().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

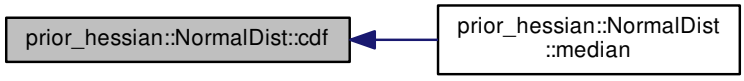
7.49.4.2 double prior_hessian::NormalDist::cdf (double *x*) const

Definition at line 42 of file NormalDist.cpp.

References prior_hessian::constants::sqrt2_inv.

Referenced by median().

Here is the caller graph for this function:



7.49.4.3 void prior_hessian::UnivariateDist::check_bounds (double *lbound,* double *ubound*) [static],[protected], [inherited]

Definition at line 17 of file UnivariateDist.cpp.

Referenced by prior_hessian::UnivariateDist::UnivariateDist().

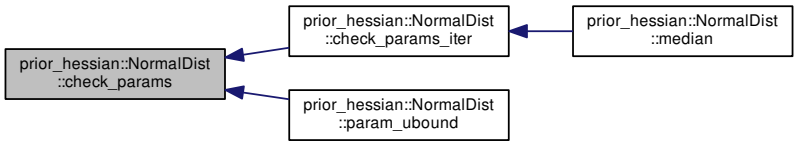
Here is the caller graph for this function:

7.49.4.4 bool prior_hessian::NormalDist::check_params (double *mu,* double *sigma*) [inline],[static]

Definition at line 121 of file NormalDist.h.

Referenced by check_params_iter(), and param_ubound().

Here is the caller graph for this function:



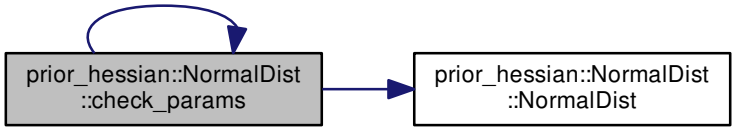
7.49.4.5 template<class Vec > static bool prior_hessian::NormalDist::check_params (const Vec & *p*) [inline], [static]

Definition at line 39 of file NormalDist.h.

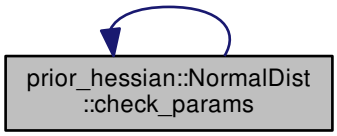
References check_params(), and NormalDist().

Referenced by check_params().

Here is the call graph for this function:



Here is the caller graph for this function:



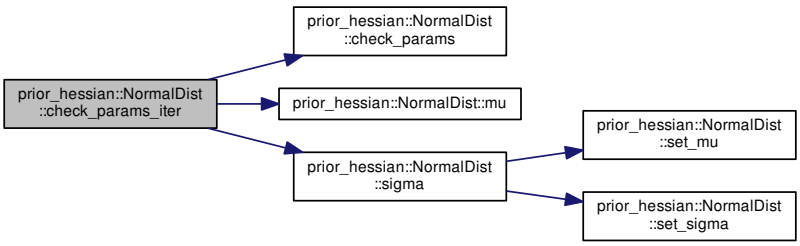
7.49.4.6 template<class IterT > bool prior_hessian::NormalDist::check_params_iter (IterT & *params*) [static]

Definition at line 197 of file NormalDist.h.

References check_params(), mu(), and sigma().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

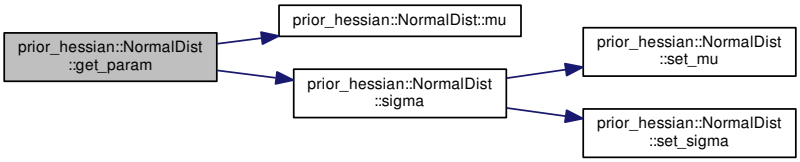
7.49.4.7 double prior_hessian::NormalDist::get_param (int *idx*) const [inline]

Definition at line 127 of file NormalDist.h.

References mu(), and sigma().

Referenced by operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

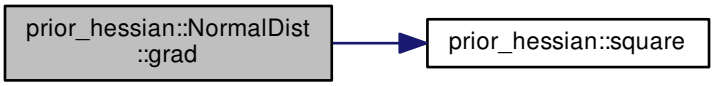
7.49.4.8 double prior_hessian::NormalDist::grad (double *x*) const [inline]

Definition at line 168 of file NormalDist.h.

References prior_hessian::square().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

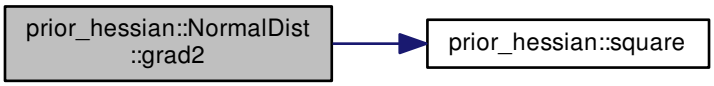
7.49.4.9 double prior_hessian::NormalDist::grad2 (double *x*) const [inline]

Definition at line 174 of file NormalDist.h.

References prior_hessian::square().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

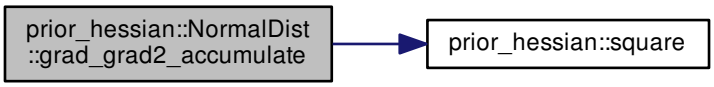
7.49.4.10 void prior_hessian::NormalDist::grad_grad2_accumulate (double *x*, double & *g*, double & *g2*) const [inline]

Definition at line 180 of file NormalDist.h.

References prior_hessian::square().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

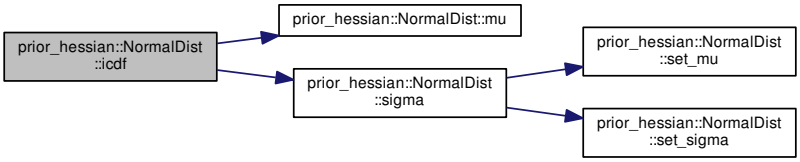
7.49.4.11 double prior_hessian::NormalDist::icdf (double *u*) const

Definition at line 47 of file NormalDist.cpp.

References mu(), sigma(), and prior_hessian::constants::sqrt2.

Referenced by median().

Here is the call graph for this function:



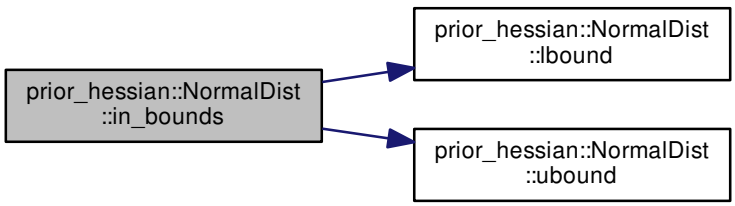
Here is the caller graph for this function:

7.49.4.12 static bool prior_hessian::NormalDist::in_bounds (double *u*) [inline], [static]

Definition at line 31 of file NormalDist.h.

References lbound(), and ubound().

Here is the call graph for this function:



7.49.4.13 static constexpr double prior_hessian::NormalDist::lbound ( ) [inline], [static]

Definition at line 29 of file NormalDist.h.

Referenced by in_bounds().

Here is the caller graph for this function:

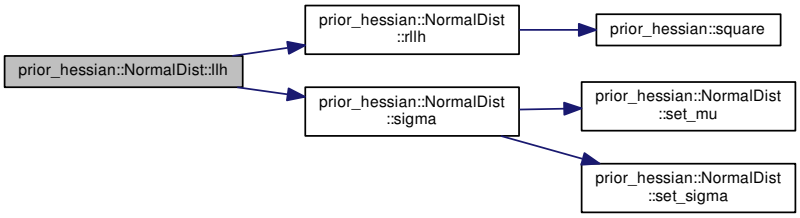
7.49.4.14 double prior_hessian::NormalDist::llh (double *x*) const

Definition at line 72 of file NormalDist.cpp.

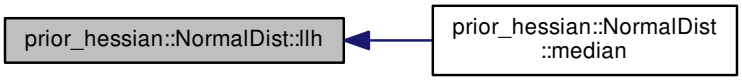
References prior_hessian::constants::log2pi, rllh(), and sigma().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

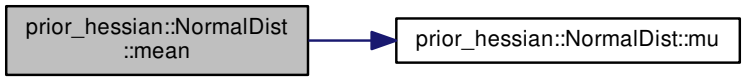


7.49.4.15 double prior_hessian::NormalDist::mean ( ) const [inline]

Definition at line 62 of file NormalDist.h.

References mu().

Here is the call graph for this function:

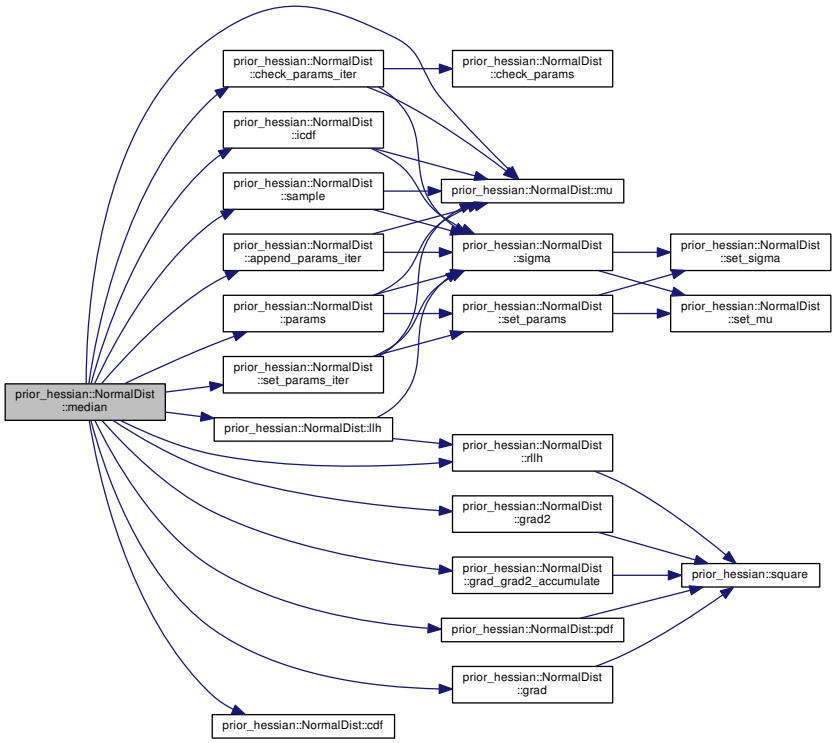


7.49.4.16 double prior_hessian::NormalDist::median ( ) const [inline]

Definition at line 63 of file NormalDist.h.

References append_params_iter(), cdf(), check_params_iter(), grad(), grad2(), grad_grad2_accumulate(), icdf(), llh(), mu(), params(), pdf(), rllh(), sample(), and set_params_iter().

Here is the call graph for this function:

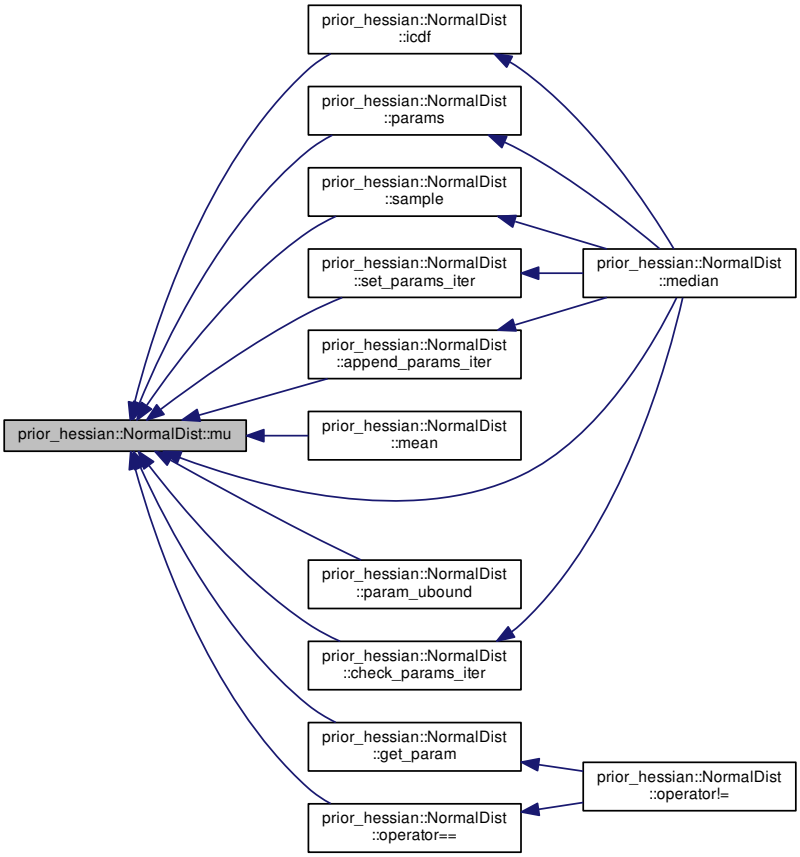


7.49.4.17 double prior_hessian::NormalDist::mu ( ) const [inline]

Definition at line 48 of file NormalDist.h.

Referenced by append_params_iter(), check_params_iter(), get_param(), icdf(), mean(), median(), operator==(), param_ubound(), params(), sample(), and set_params_iter().

Here is the caller graph for this function:



7.49.4.18 static constexpr IdxT prior_hessian::UnivariateDist::num_dim ( ) [inline], [static], [inherited]

Definition at line 17 of file UnivariateDist.h.

7.49.4.19 static constexpr IdxT prior_hessian::NormalDist::num_params ( ) [inline], [static]

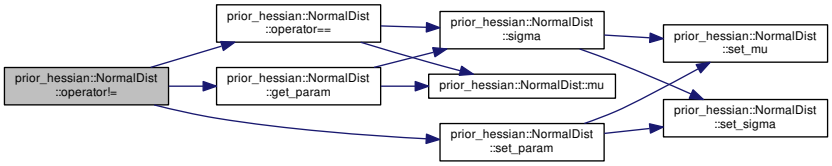
Definition at line 28 of file NormalDist.h.

7.49.4.20 bool prior_hessian::NormalDist::operator!= (const NormalDist & *o*) const [inline]

Definition at line 53 of file NormalDist.h.

References get_param(), operator==(), and set_param().

Here is the call graph for this function:



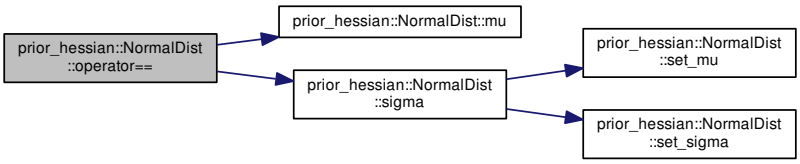
7.49.4.21 bool prior_hessian::NormalDist::operator== (const NormalDist & *o*) const [inline]

Definition at line 52 of file NormalDist.h.

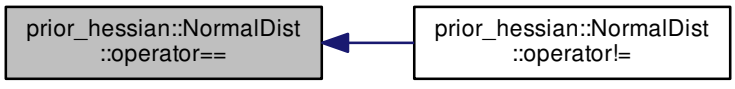
References mu(), and sigma().

Referenced by operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:



7.49.4.22 static const NparamsVecT& prior_hessian::NormalDist::param_lbound ( ) [inline], [static]

Definition at line 34 of file NormalDist.h.

7.49.4.23 static const StringVecT& prior_hessian::NormalDist::param_names ( ) [inline], [static]

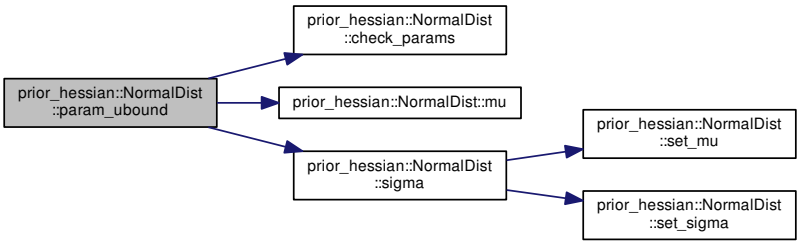
Definition at line 33 of file NormalDist.h.

7.49.4.24 static const NparamsVecT& prior_hessian::NormalDist::param_ubound ( ) [inline], [static]

Definition at line 35 of file NormalDist.h.

References check_params(), mu(), and sigma().

Here is the call graph for this function:



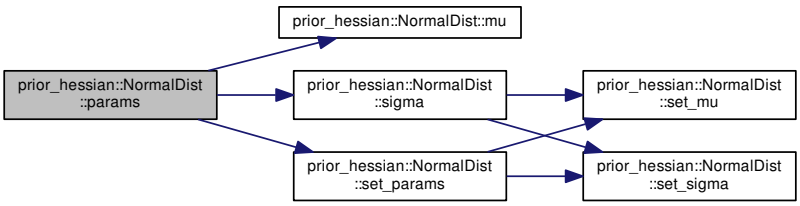
7.49.4.25 NparamsVecT prior_hessian::NormalDist::params ( ) const [inline]

Definition at line 57 of file NormalDist.h.

References mu(), set_params(), and sigma().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

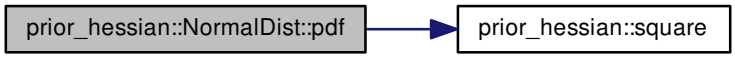
7.49.4.26 double prior_hessian::NormalDist::pdf (double *x*) const [inline]

Definition at line 156 of file NormalDist.h.

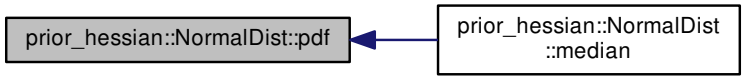
References prior_hessian::constants::sqrt2pi_inv, and prior_hessian::square().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:



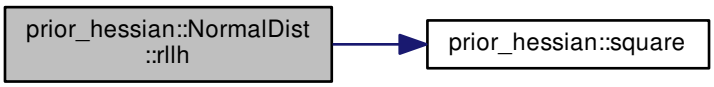
7.49.4.27 double prior_hessian::NormalDist::rllh (double *x*) const [inline]

Definition at line 162 of file NormalDist.h.

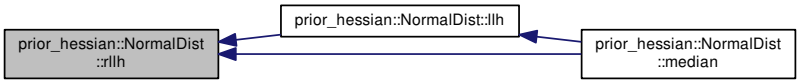
References prior_hessian::square().

Referenced by llh(), and median().

Here is the call graph for this function:



Here is the caller graph for this function:



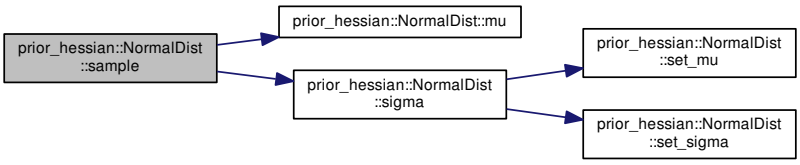
7.49.4.28 template<class RngT > double prior_hessian::NormalDist::sample (RngT & *rng*) const

Definition at line 188 of file NormalDist.h.

References mu(), and sigma().

Referenced by median().

Here is the call graph for this function:



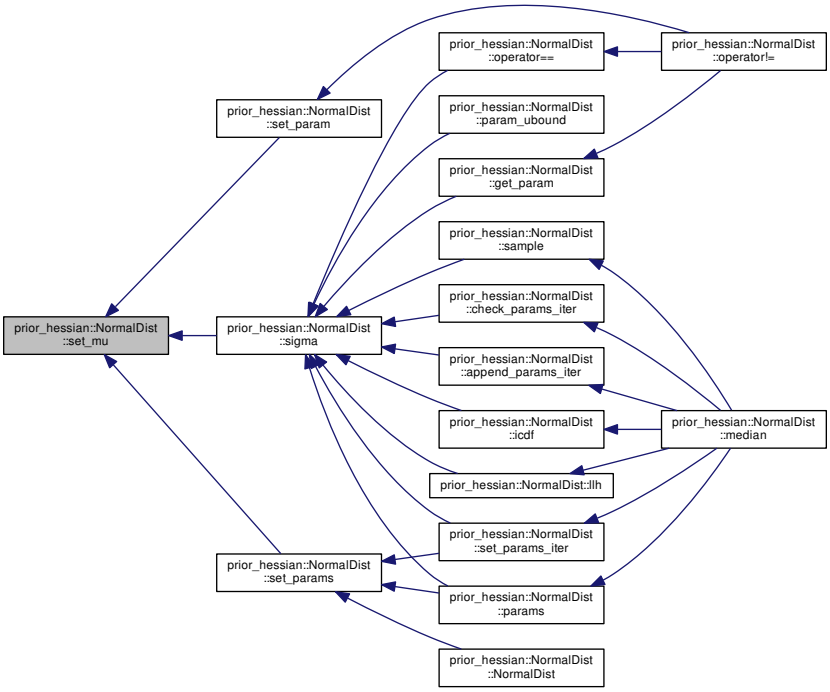
Here is the caller graph for this function:

7.49.4.29 void prior_hessian::NormalDist::set_mu (double *val*) [inline]

Definition at line 110 of file NormalDist.h.

Referenced by set_param(), set_params(), and sigma().

Here is the caller graph for this function:



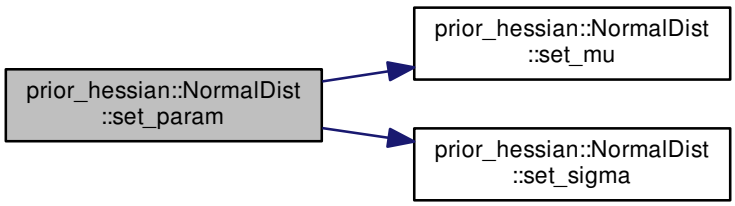
7.49.4.30 void prior_hessian::NormalDist::set_param (int *idx,* double *val*) [inline]

Definition at line 141 of file NormalDist.h.

References set_mu(), and set_sigma().

Referenced by operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

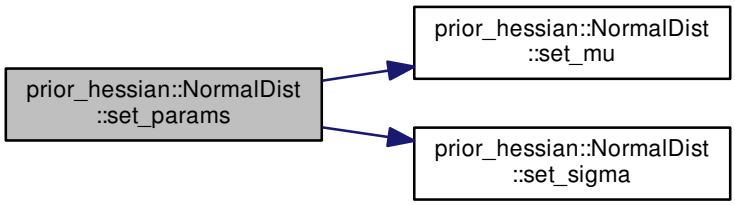
7.49.4.31 void prior_hessian::NormalDist::set_params (double *mu,* double *sigma*) [inline]

Definition at line 114 of file NormalDist.h.

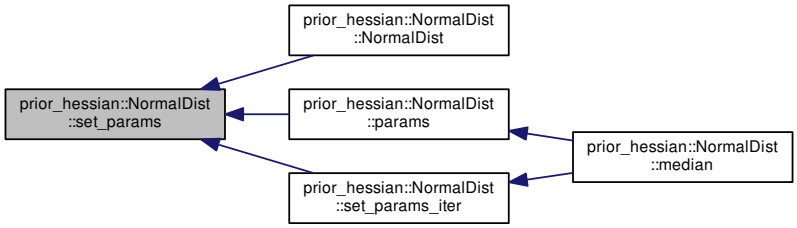
References set_mu(), and set_sigma().

Referenced by NormalDist(), params(), and set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:



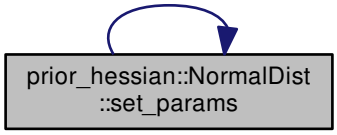
7.49.4.32 template<class Vec > void prior_hessian::NormalDist::set_params (const Vec & *p*) [inline]

Definition at line 60 of file NormalDist.h.

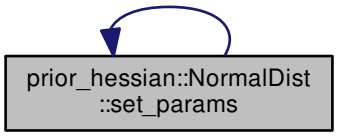
References set_params().

Referenced by set_params().

Here is the call graph for this function:



Here is the caller graph for this function:



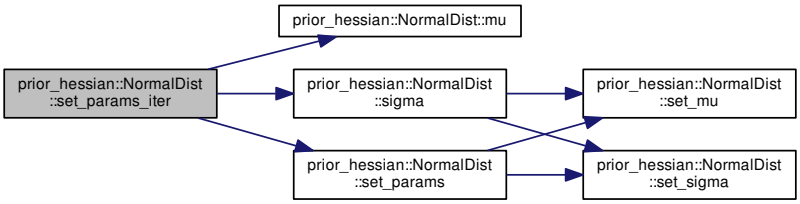
7.49.4.33 template<class IterT > void prior_hessian::NormalDist::set_params_iter (IterT & *params*)

Definition at line 213 of file NormalDist.h.

References mu(), set_params(), and sigma().

Referenced by median().

Here is the call graph for this function:



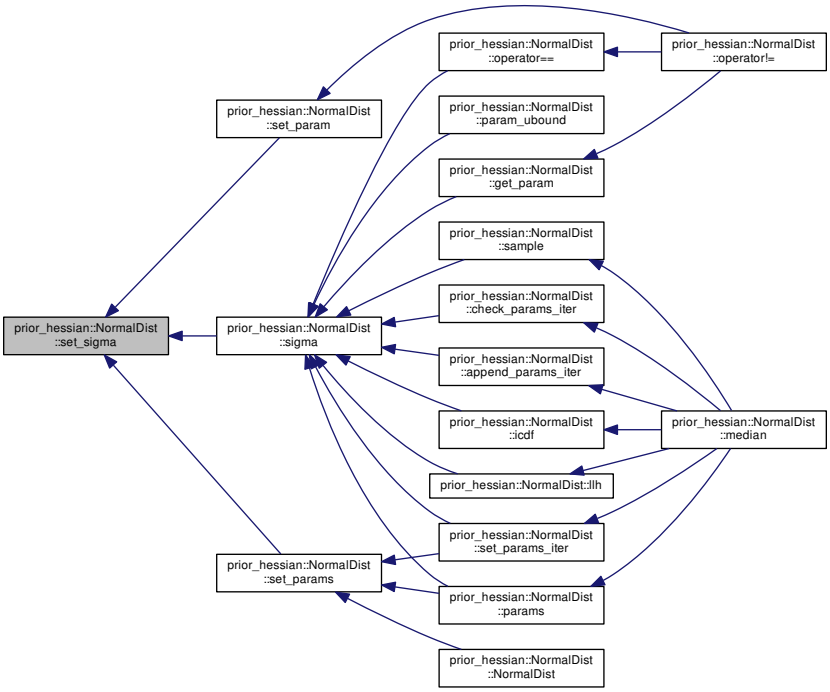
Here is the caller graph for this function:

7.49.4.34 void prior_hessian::NormalDist::set_sigma (double *val*)

Definition at line 35 of file NormalDist.cpp.

Referenced by set_param(), set_params(), and sigma().

Here is the caller graph for this function:



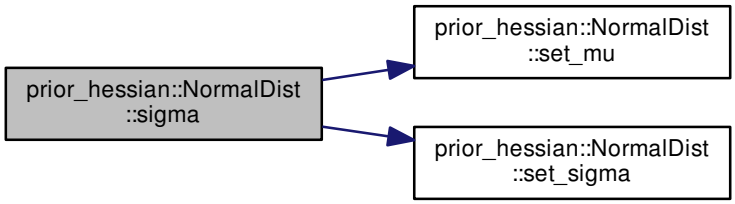
7.49.4.35 double prior_hessian::NormalDist::sigma ( ) const [inline]

Definition at line 49 of file NormalDist.h.

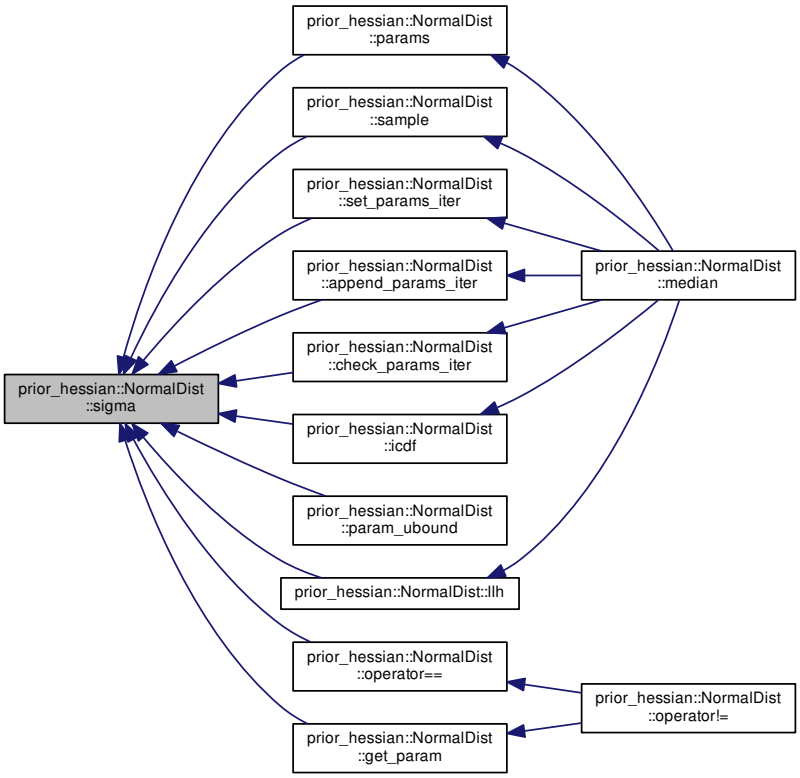
References set_mu(), and set_sigma().

Referenced by append_params_iter(), check_params_iter(), get_param(), icdf(), llh(), operator==(), param_ubound(), params(), sample(), and set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:



7.49.4.36 static constexpr double prior_hessian::NormalDist::ubound ( ) [inline],[static]

Definition at line 30 of file NormalDist.h.

Referenced by in_bounds().

Here is the caller graph for this function:

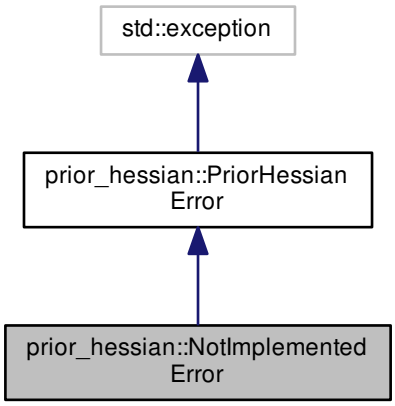
The documentation for this class was generated from the following files:

- NormalDist.h
- NormalDist.cpp

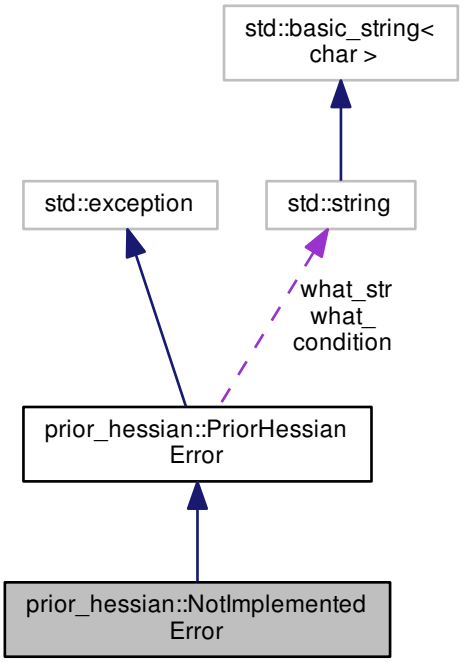
7.50 prior_hessian::NotImplementedError Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::NotImplementedError:



Collaboration diagram for prior_hessian::NotImplementedError:



Public Member Functions

- NotImplementedError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.50.1 Detailed Description

Definition at line 114 of file PriorHessianError.h.

7.50.2 Constructor & Destructor Documentation

7.50.2.1 prior_hessian::NotImplementedError::NotImplementedError (std::string *message*) [inline]

Definition at line 116 of file PriorHessianError.h.

7.50.3 Member Function Documentation

7.50.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.50.4 Member Data Documentation

7.50.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.50.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.50.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

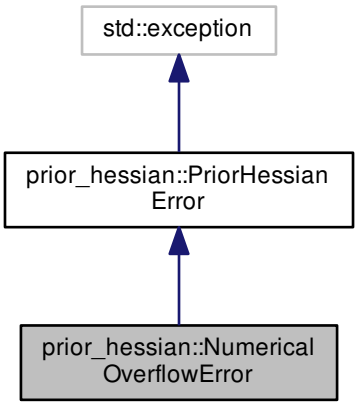
The documentation for this struct was generated from the following file:

- PriorHessianError.h

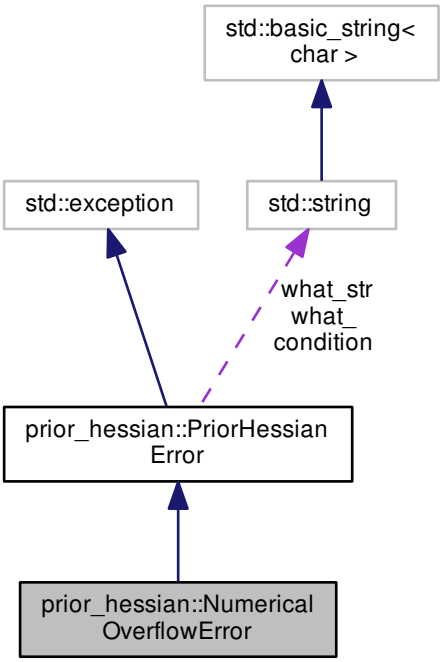
7.51 prior_hessian::NumericalOverflowError Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::NumericalOverflowError:



Collaboration diagram for prior_hessian::NumericalOverflowError:



Public Member Functions

- NumericalOverflowError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.51.1 Detailed Description

Definition at line 109 of file PriorHessianError.h.

7.51.2 Constructor & Destructor Documentation

7.51.2.1 prior_hessian::NumericalOverflowError::NumericalOverflowError (std::string *message*) [inline]

Definition at line 111 of file PriorHessianError.h.

7.51.3 Member Function Documentation

7.51.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.51.4 Member Data Documentation

7.51.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.51.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.51.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

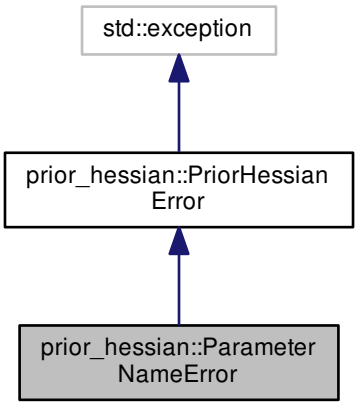
The documentation for this struct was generated from the following file:

- PriorHessianError.h

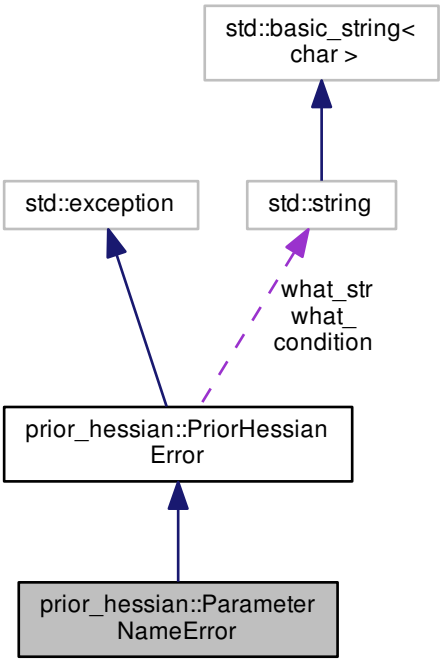
7.52 prior_hessian::ParameterNameError Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::ParameterNameError:



Collaboration diagram for prior_hessian::ParameterNameError:



Public Member Functions

- ParameterNameError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.52.1 Detailed Description

Definition at line 85 of file PriorHessianError.h.

7.52.2 Constructor & Destructor Documentation

7.52.2.1 prior_hessian::ParameterNameError::ParameterNameError (std::string *message*) [inline]

Definition at line 87 of file PriorHessianError.h.

7.52.3 Member Function Documentation

7.52.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.52.4 Member Data Documentation

7.52.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.52.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.52.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

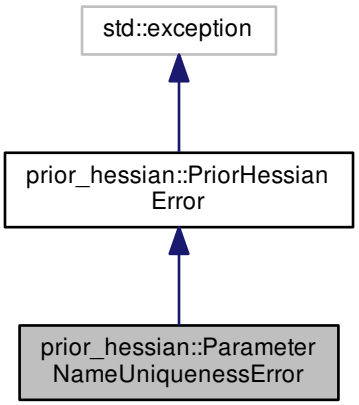
The documentation for this struct was generated from the following file:

- PriorHessianError.h

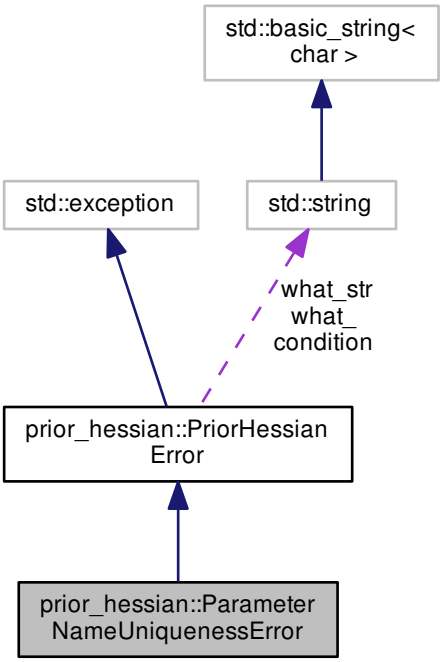
7.53 prior_hessian::ParameterNameUniquenessError Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::ParameterNameUniquenessError:



Collaboration diagram for prior_hessian::ParameterNameUniquenessError:



Public Member Functions

- ParameterNameUniquenessError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.53.1 Detailed Description

Definition at line 89 of file PriorHessianError.h.

7.53.2 Constructor & Destructor Documentation

7.53.2.1 prior_hessian::ParameterNameUniquenessError::ParameterNameUniquenessError (std::string *message*) [inline]

Definition at line 91 of file PriorHessianError.h.

7.53.3 Member Function Documentation

7.53.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.53.4 Member Data Documentation

7.53.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.53.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.53.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

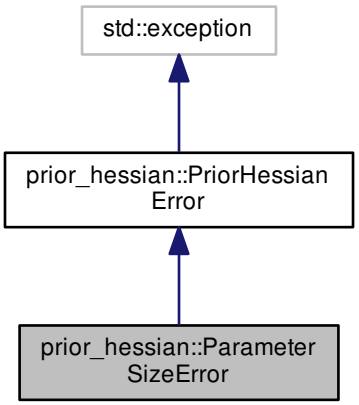
The documentation for this struct was generated from the following file:

- PriorHessianError.h

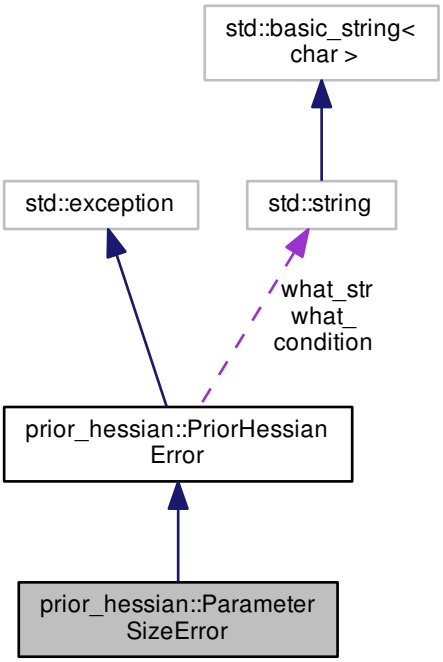
7.54 prior_hessian::ParameterSizeError Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::ParameterSizeError:



Collaboration diagram for prior_hessian::ParameterSizeError:



Public Member Functions

- ParameterSizeError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.54.1 Detailed Description

Definition at line 75 of file PriorHessianError.h.

7.54.2 Constructor & Destructor Documentation

7.54.2.1 prior_hessian::ParameterSizeError::ParameterSizeError (std::string *message*) [inline]

Definition at line 77 of file PriorHessianError.h.

7.54.3 Member Function Documentation

7.54.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.54.4 Member Data Documentation

7.54.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.54.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.54.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

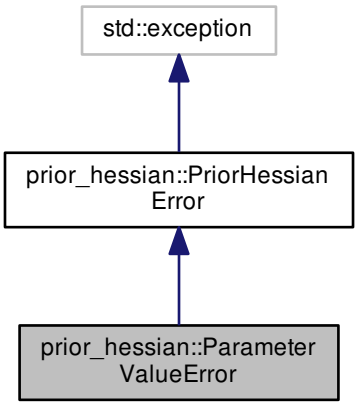
The documentation for this struct was generated from the following file:

- PriorHessianError.h

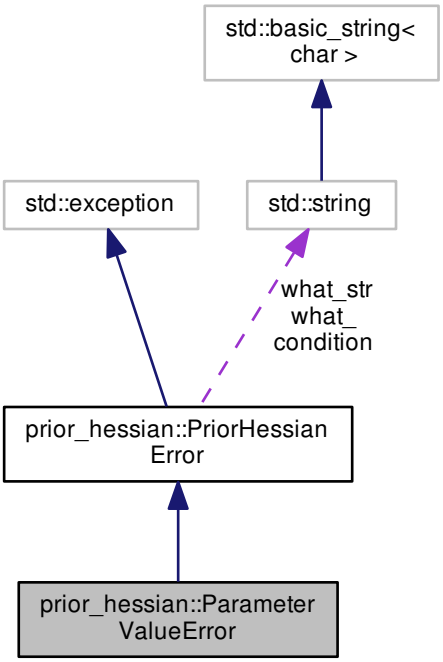
7.55 prior_hessian::ParameterValueError Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::ParameterValueError:



Collaboration diagram for prior_hessian::ParameterValueError:



Public Member Functions

- ParameterValueError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.55.1 Detailed Description

Definition at line 80 of file PriorHessianError.h.

7.55.2 Constructor & Destructor Documentation

7.55.2.1 prior_hessian::ParameterValueError::ParameterValueError (std::string *message*) [inline]

Definition at line 82 of file PriorHessianError.h.

7.55.3 Member Function Documentation

7.55.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.55.4 Member Data Documentation

7.55.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.55.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.55.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

The documentation for this struct was generated from the following file:

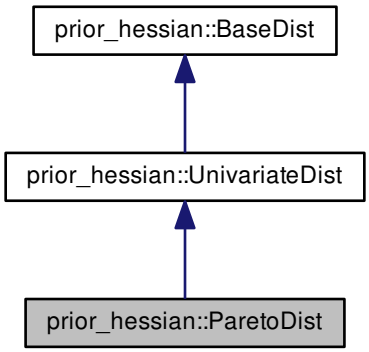
- PriorHessianError.h

7.56 prior_hessian::ParetoDist Class Reference

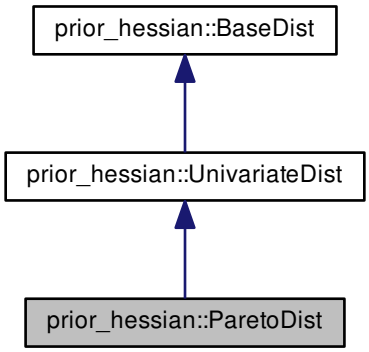
Pareto dist with infinite upper bound.

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/ParetoDist.h>`

Inheritance diagram for prior_hessian::ParetoDist:



Collaboration diagram for prior_hessian::ParetoDist:



Public Types

- using NparamsVecT = arma::Col< double >::fixed< _num_params >

Public Member Functions

- bool in_bounds (double u) const
- ParetoDist ()
- ParetoDist (double min, double alpha)
- template<class Vec , meta::ConstructableIfNotSelfT< Vec, ParetoDist > = true>
  ParetoDist (const Vec &params)
- double get_param (int idx) const
- void set_param (int idx, double val)
- NparamsVecT params () const
- void set_params (double min, double alpha)
- template<class Vec >
  void set_params (const Vec &p)
- bool operator== (const ParetoDist &o) const
- bool operator!= (const ParetoDist &o) const
- double alpha () const
- double min () const
- void set_min (double val)
- void set_alpha (double val)
- double lbound () const
- void set_lbound (double lbound)
- double mean () const
- double median () const
- double cdf (double x) const
- double icdf (double u) const

- double pdf (double x) const
- double llh (double x) const
- double rllh (double x) const
- double grad (double x) const
- double grad2 (double x) const
- void grad_grad2_accumulate (double x, double &g, double &g2) const
- template<class RngT >
  double sample (RngT &rng) const
- template<class IterT >
  void set_params_iter (IterT &params)

Static Public Member Functions

- static constexpr IdxT num_params ()
- static constexpr double global_lbound ()
- static constexpr double ubound ()
- static const StringVecT & param_names ()
- static const NparamsVecT & param_lbound ()
- static const NparamsVecT & param_ubound ()
- static bool check_params (double min, double alpha)
- template<class Vec >
  static bool check_params (const Vec &params)
- static bool check_lbound (double min)
- template<class IterT >
  static bool check_params_iter (IterT &params)
- static constexpr IdxT num_dim ()

Static Protected Member Functions

- static void check_bounds (double lbound, double ubound)

7.56.1 Detailed Description

Pareto dist with infinite upper bound.

Definition at line 20 of file ParetoDist.h.

7.56.2 Member Typedef Documentation

7.56.2.1 using prior_hessian::ParetoDist::NparamsVecT = arma::Col<double>::fixed<_num_params>

Definition at line 24 of file ParetoDist.h.

7.56.3 Constructor & Destructor Documentation

7.56.3.1 prior_hessian::ParetoDist::ParetoDist ( ) [inline]

Definition at line 41 of file ParetoDist.h.

7.56.3.2 prior_hessian::ParetoDist::ParetoDist (double *min,* double *alpha*)

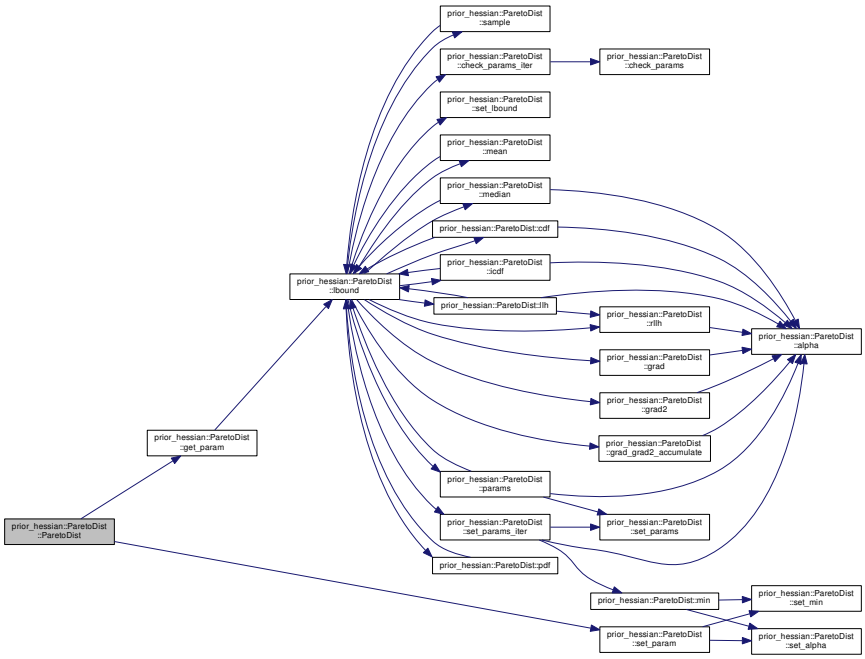
Definition at line 21 of file ParetoDist.cpp.

7.56.3.3 template<class Vec , meta::ConstructableIfNotSelfT< Vec, ParetoDist > = true> prior_hessian::ParetoDist::ParetoDist (const Vec & *params*) [inline], [explicit]

Definition at line 44 of file ParetoDist.h.

References get_param(), and set_param().

Here is the call graph for this function:



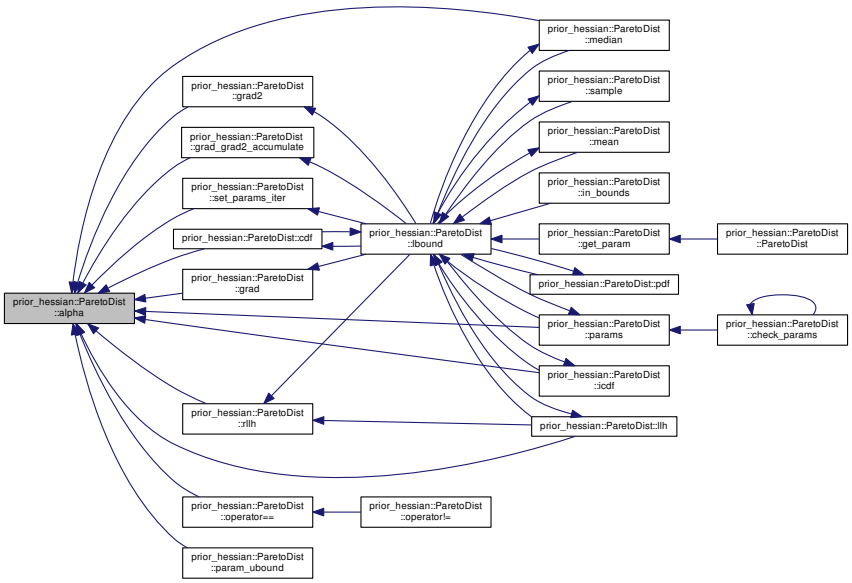
7.56.4 Member Function Documentation

7.56.4.1 double prior_hessian::ParetoDist::alpha ( ) const [inline]

Definition at line 55 of file ParetoDist.h.

Referenced by cdf(), grad(), grad2(), grad_grad2_accumulate(), icdf(), llh(), median(), operator==(), param_ubound(), params(), rllh(), and set_params_iter().

Here is the caller graph for this function:



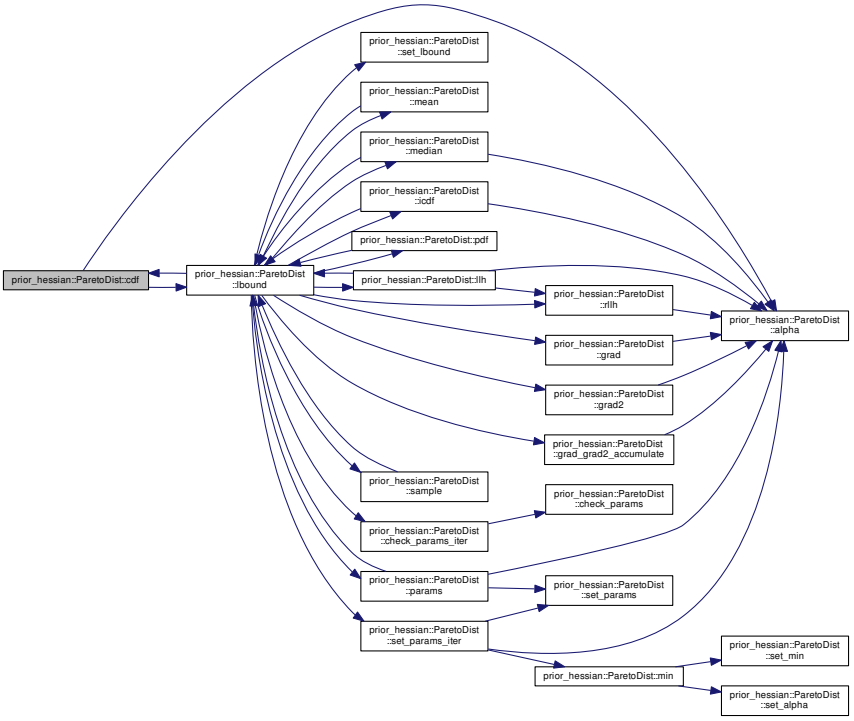
7.56.4.2 double prior_hessian::ParetoDist::cdf (double *x*) const [inline]

Definition at line 155 of file ParetoDist.h.

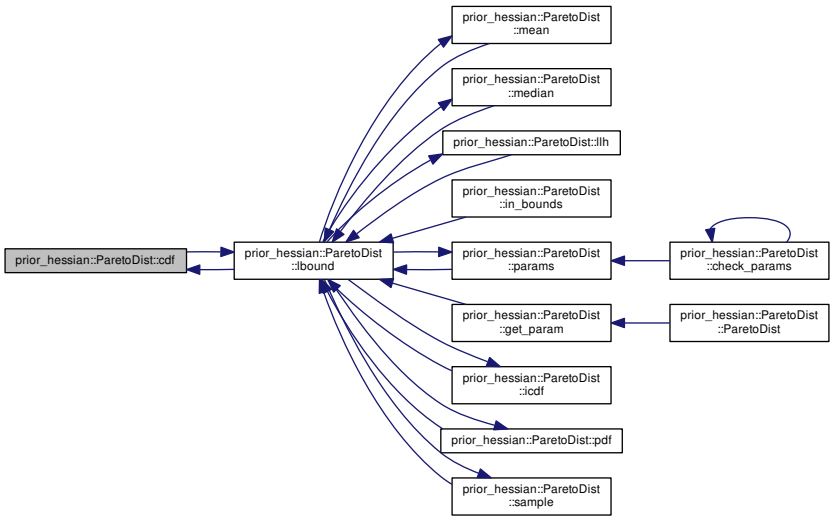
References alpha(), and lbound().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



7.56.4.3 void prior_hessian::UnivariateDist::check_bounds (double *lbound,* double *ubound*) [static],[protected], [inherited]

Definition at line 17 of file UnivariateDist.cpp.

Referenced by prior_hessian::UnivariateDist::UnivariateDist().

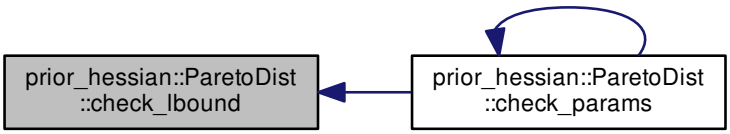
Here is the caller graph for this function:

7.56.4.4 bool prior_hessian::ParetoDist::check_lbound (double *min*) [inline],[static]

Definition at line 149 of file ParetoDist.h.

Referenced by check_params().

Here is the caller graph for this function:

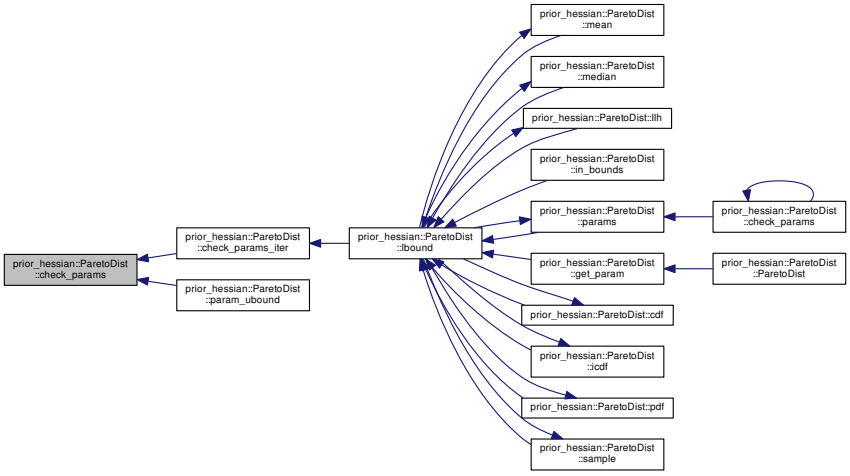


7.56.4.5 bool prior_hessian::ParetoDist::check_params (double *min,* double *alpha*) [inline],[static]

Definition at line 104 of file ParetoDist.h.

Referenced by check_params_iter(), and param_ubound().

Here is the caller graph for this function:



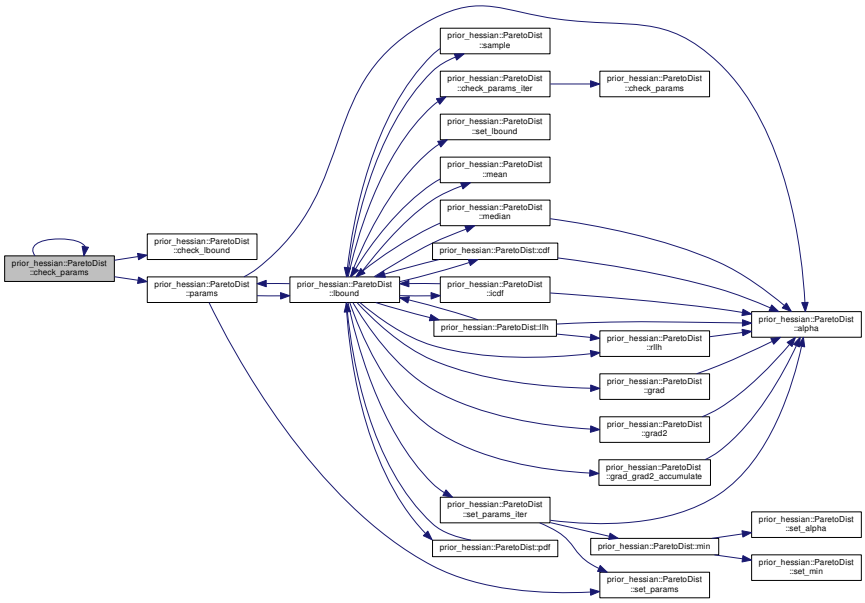
7.56.4.6 template<class Vec > static bool prior_hessian::ParetoDist::check_params (const Vec & *params*) [inline], [static]

Definition at line 38 of file ParetoDist.h.

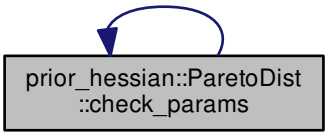
References check_lbound(), check_params(), and params().

Referenced by check_params().

Here is the call graph for this function:



Here is the caller graph for this function:



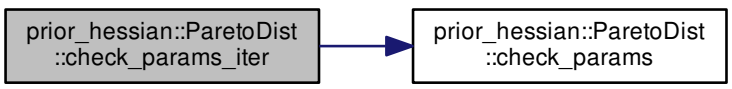
7.56.4.7 template<class IterT > bool prior_hessian::ParetoDist::check_params_iter (IterT & *params*) [static]

Definition at line 111 of file ParetoDist.h.

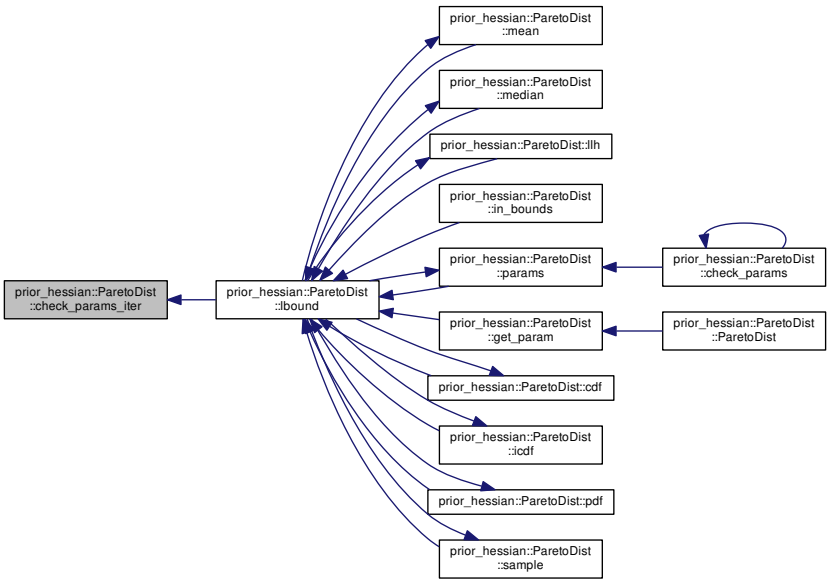
References check_params().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



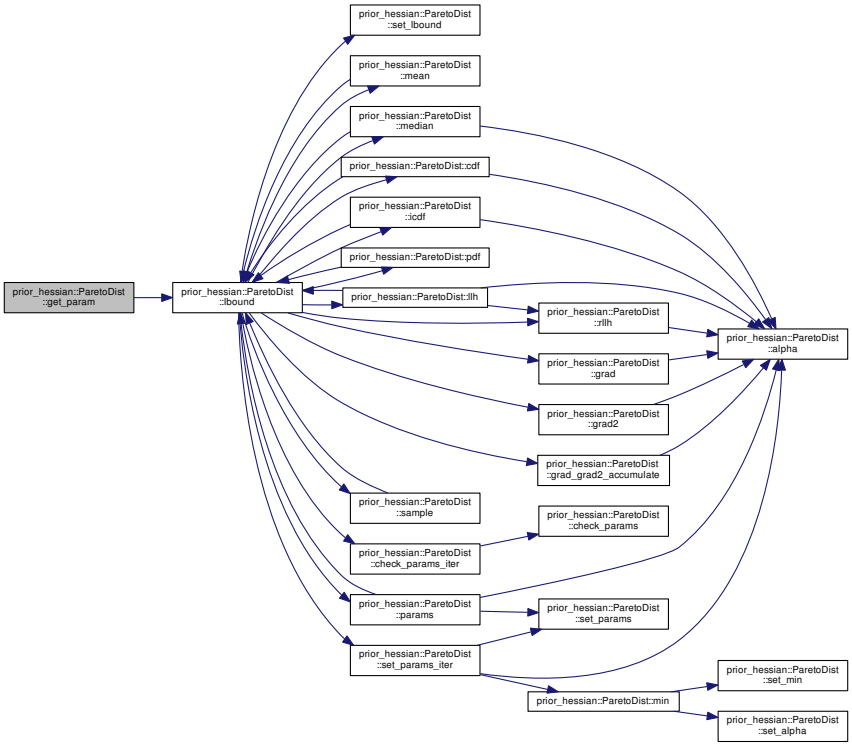
7.56.4.8 double prior_hessian::ParetoDist::get_param (int *idx*) const [inline]

Definition at line 119 of file ParetoDist.h.

References lbound().

Referenced by ParetoDist().

Here is the call graph for this function:



Here is the caller graph for this function:

7.56.4.9 static constexpr double prior_hessian::ParetoDist::global_lbound ( ) [inline], [static]

Definition at line 26 of file ParetoDist.h.

7.56.4.10 double prior_hessian::ParetoDist::grad (double *x*) const [inline]

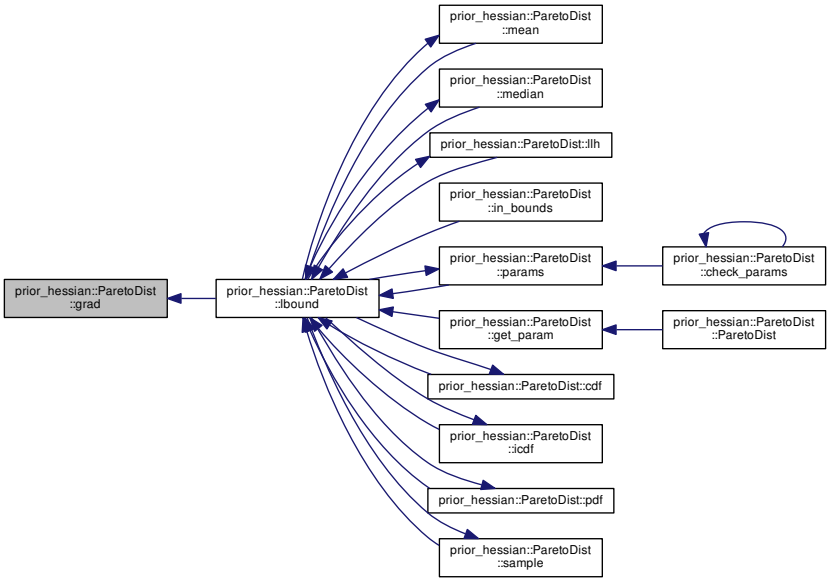
Definition at line 179 of file ParetoDist.h.

References alpha().

Referenced by lbound().

Here is the call graph for this function:

Here is the caller graph for this function:



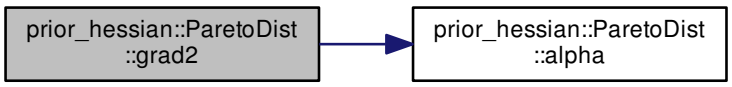
7.56.4.11 double prior_hessian::ParetoDist::grad2 (double *x*) const [inline]

Definition at line 185 of file ParetoDist.h.

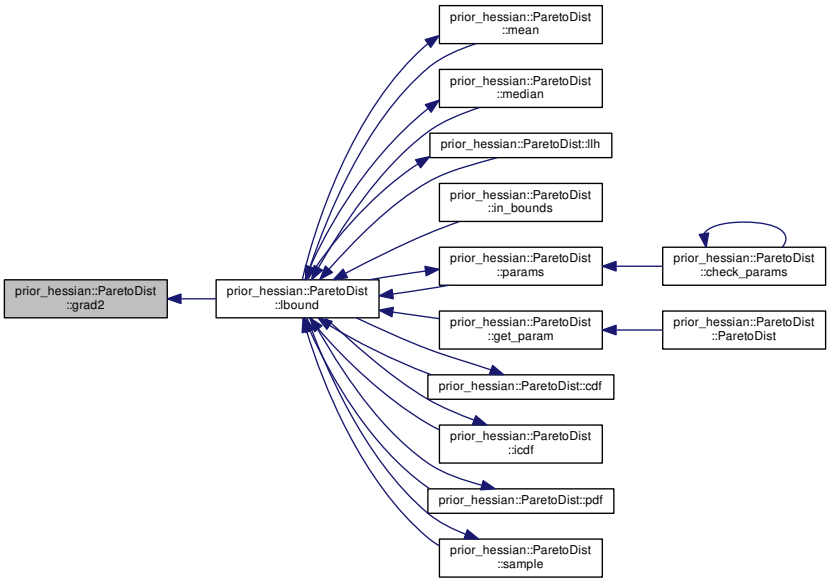
References alpha().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



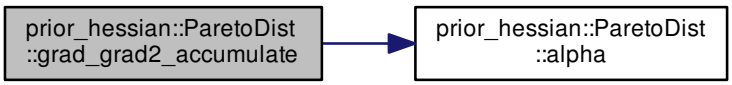
7.56.4.12 void prior_hessian::ParetoDist::grad_grad2_accumulate (double *x*, double & *g*, double & *g2*) const [inline]

Definition at line 191 of file ParetoDist.h.

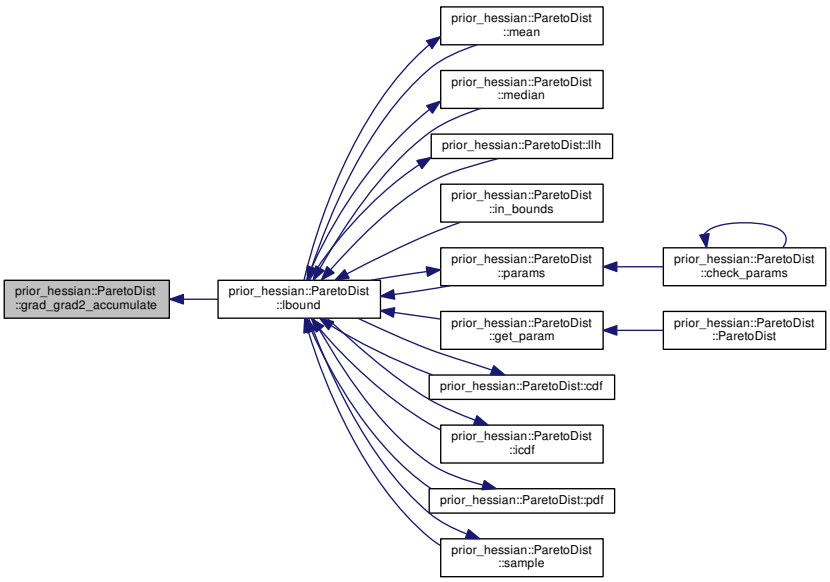
References alpha().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



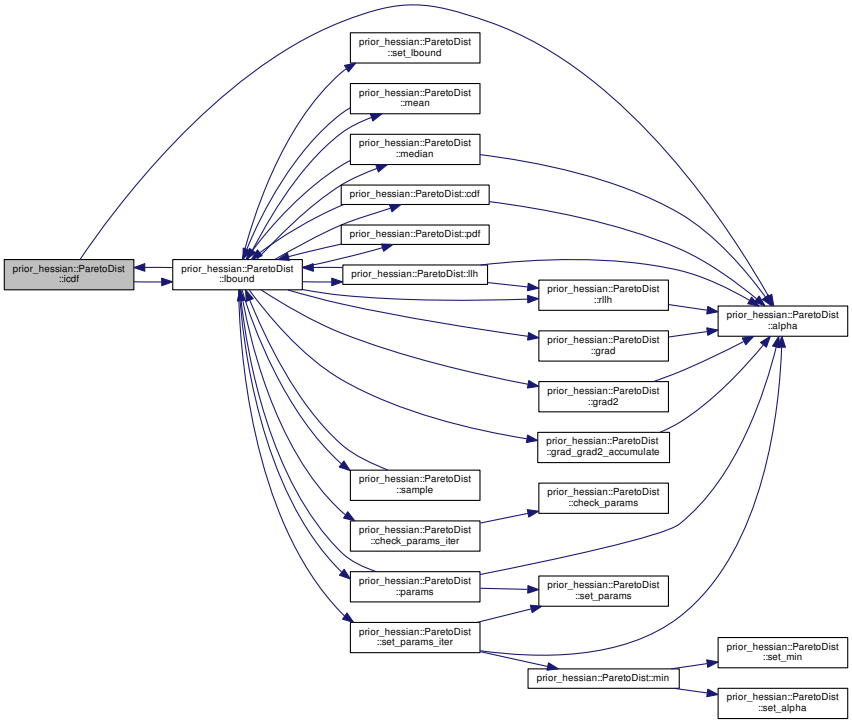
7.56.4.13 double prior_hessian::ParetoDist::icdf (double *u*) const [inline]

Definition at line 161 of file ParetoDist.h.

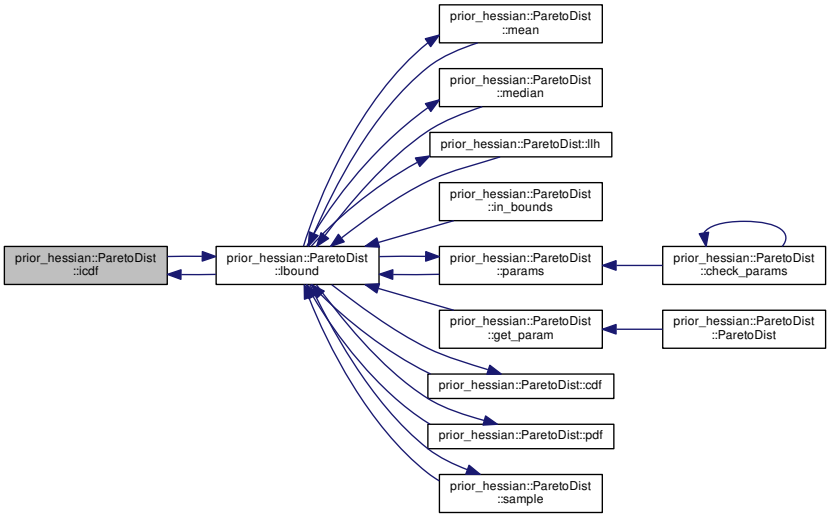
References alpha(), and lbound().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:

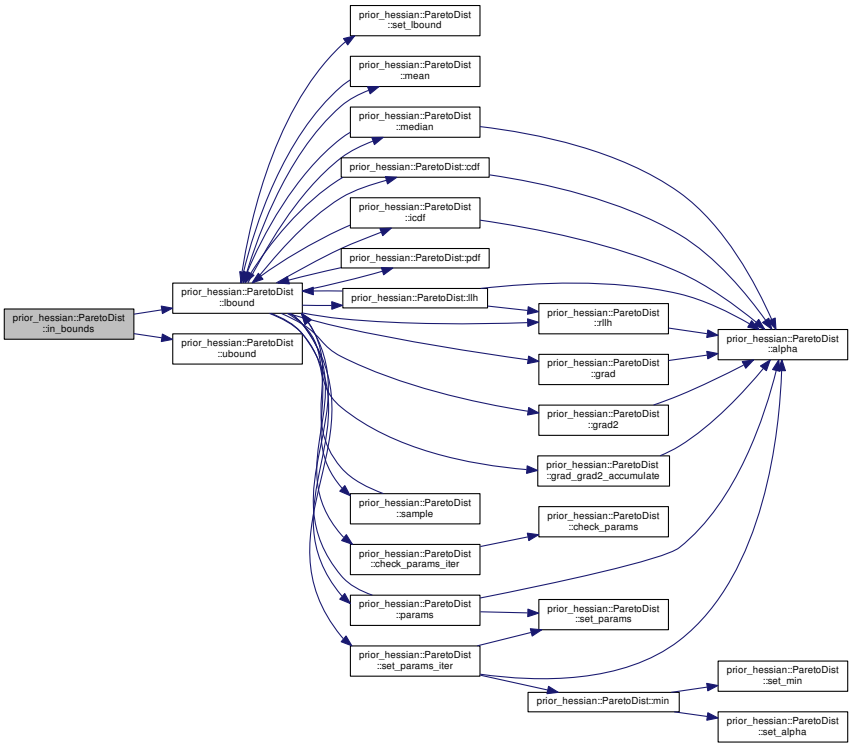


7.56.4.14 bool prior_hessian::ParetoDist::in_bounds (double *u*) const [inline]

Definition at line 28 of file ParetoDist.h.

References lbound(), and ubound().

Here is the call graph for this function:



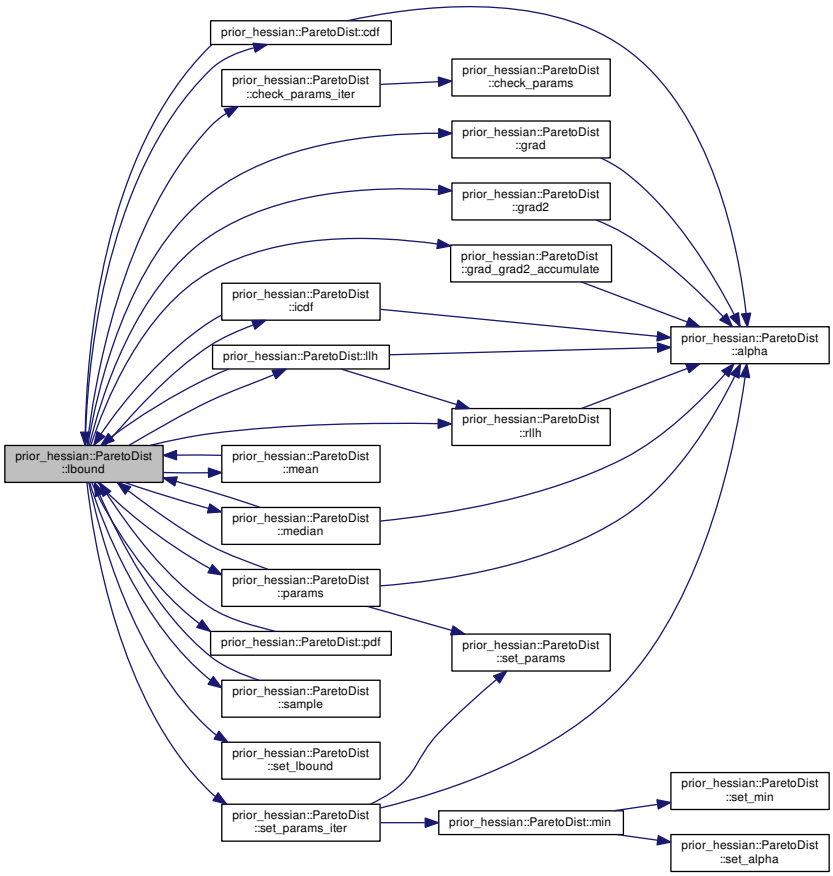
7.56.4.15 double prior_hessian::ParetoDist::lbound ( ) const [inline]

Definition at line 60 of file ParetoDist.h.

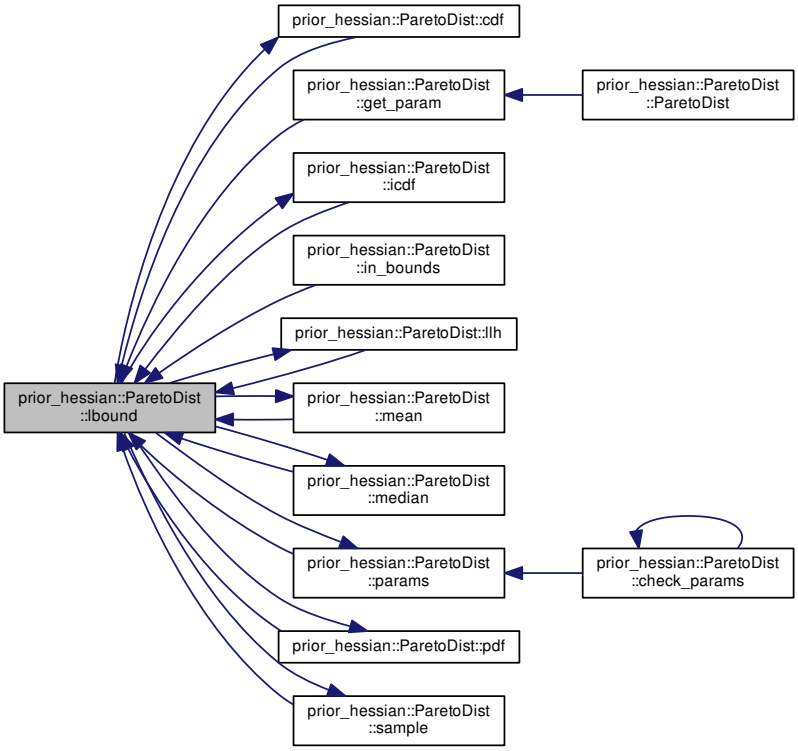
References cdf(), check_params_iter(), grad(), grad2(), grad_grad2_accumulate(), icdf(), llh(), mean(), median(), params(), pdf(), rllh(), sample(), set_lbound(), and set_params_iter().

Referenced by cdf(), get_param(), icdf(), in_bounds(), llh(), mean(), median(), params(), pdf(), and sample().

Here is the call graph for this function:



Here is the caller graph for this function:



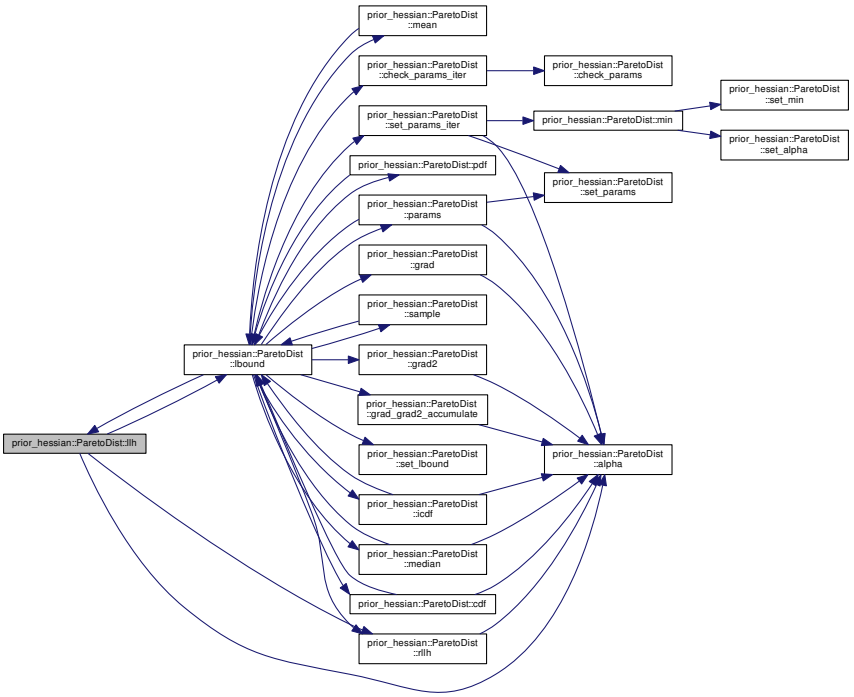
7.56.4.16 double prior_hessian::ParetoDist::llh (double *x*) const

Definition at line 64 of file ParetoDist.cpp.

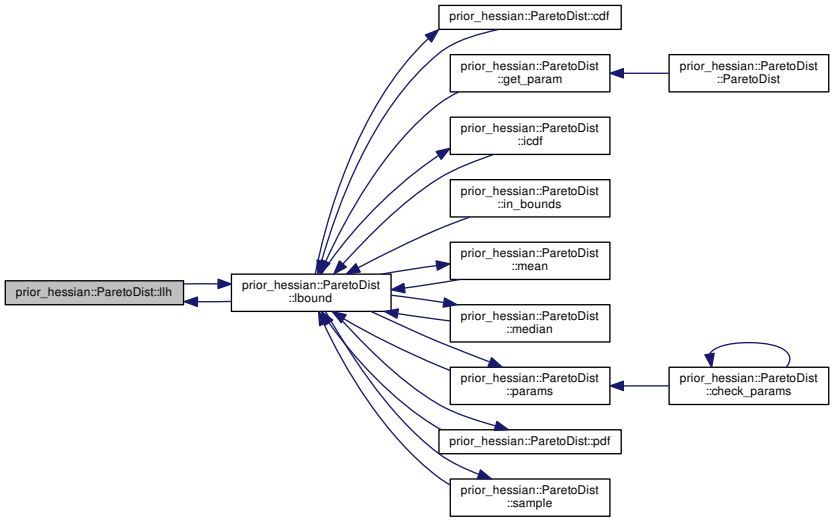
References alpha(), lbound(), and rllh().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



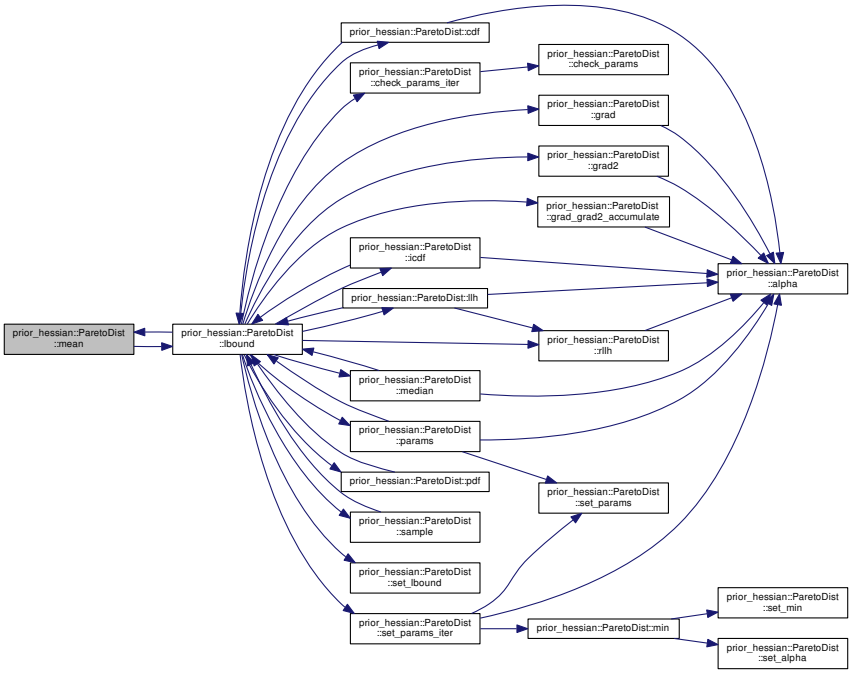
7.56.4.17 double prior_hessian::ParetoDist::mean ( ) const

Definition at line 54 of file ParetoDist.cpp.

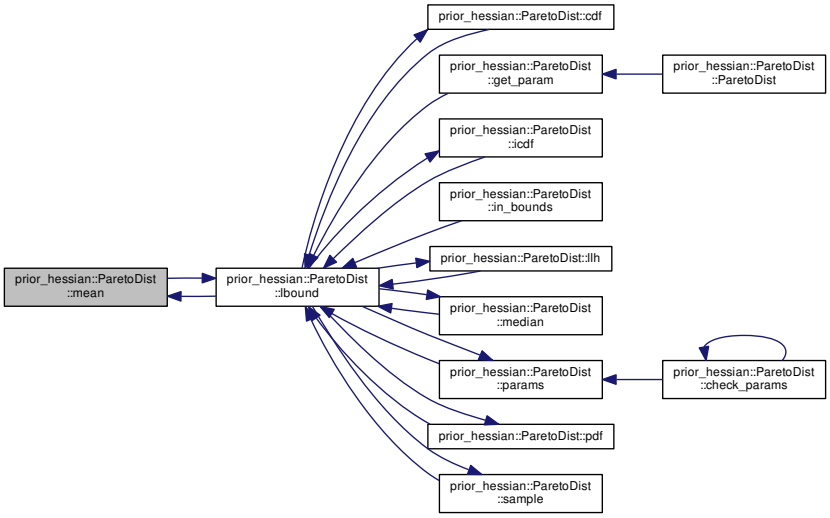
References lbound().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



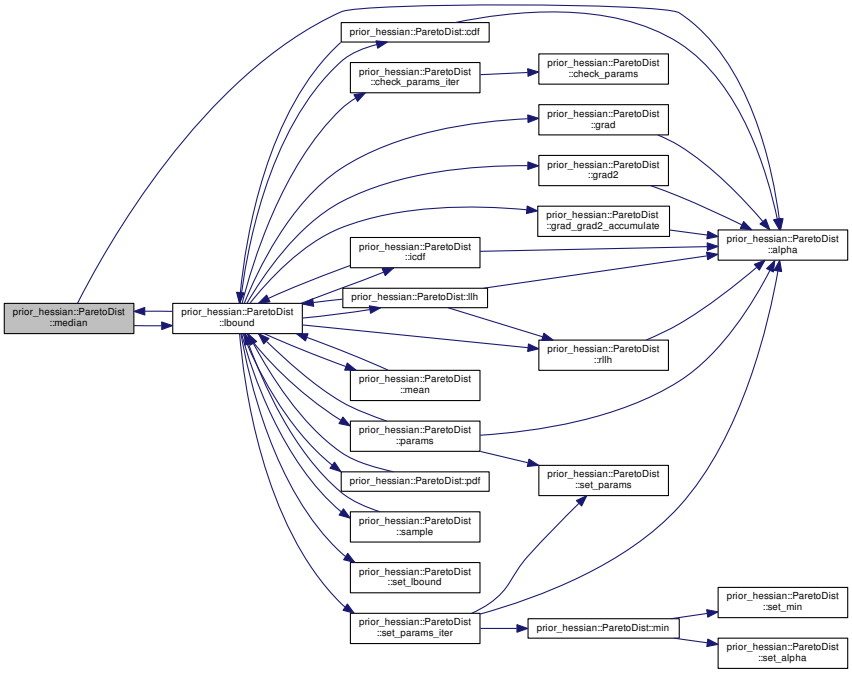
7.56.4.18 double prior_hessian::ParetoDist::median ( ) const

Definition at line 59 of file ParetoDist.cpp.

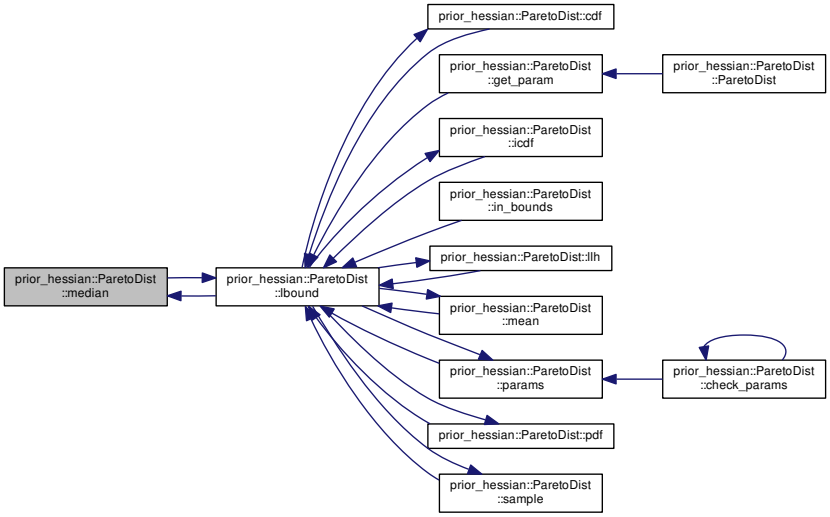
References alpha(), and lbound().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



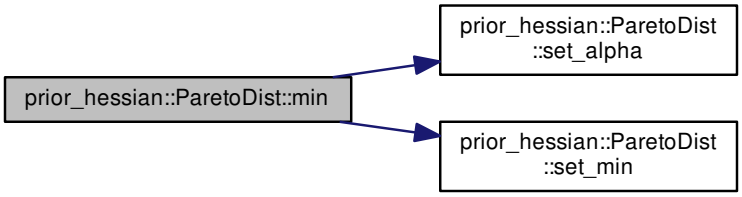
7.56.4.19 double prior_hessian::ParetoDist::min ( ) const [inline]

Definition at line 56 of file ParetoDist.h.

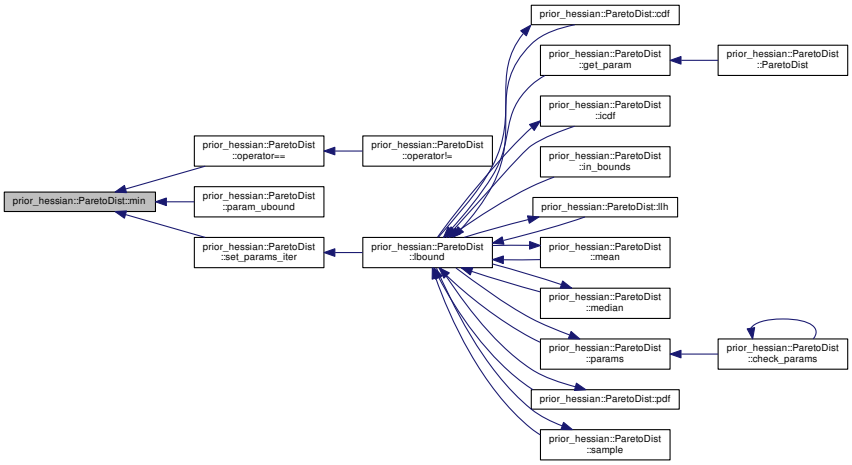
References set_alpha(), and set_min().

Referenced by operator==(), param_ubound(), and set_params_iter().

Here is the call graph for this function:



Here is the caller graph for this function:



7.56.4.20 static constexpr IdxT prior_hessian::UnivariateDist::num_dim ( ) [inline], [static], [inherited]

Definition at line 17 of file UnivariateDist.h.

7.56.4.21 static constexpr IdxT prior_hessian::ParetoDist::num_params ( ) [inline], [static]

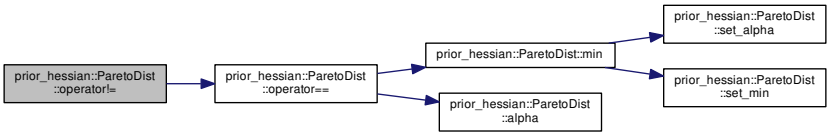
Definition at line 25 of file ParetoDist.h.

7.56.4.22 bool prior_hessian::ParetoDist::operator!= (const ParetoDist & *o*) const [inline]

Definition at line 53 of file ParetoDist.h.

References operator==().

Here is the call graph for this function:



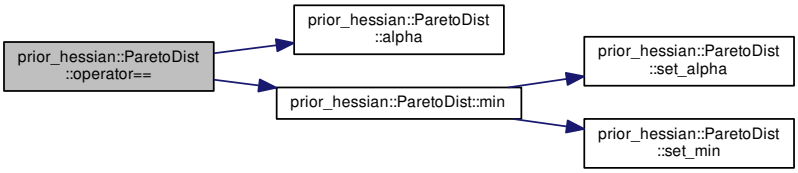
7.56.4.23 bool prior_hessian::ParetoDist::operator== (const ParetoDist & *o*) const [inline]

Definition at line 52 of file ParetoDist.h.

References alpha(), and min().

Referenced by operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

7.56.4.24 static const NparamsVecT& prior_hessian::ParetoDist::param_lbound ( ) [inline], [static]

Definition at line 32 of file ParetoDist.h.

7.56.4.25 static const StringVecT& prior_hessian::ParetoDist::param_names ( ) [inline], [static]

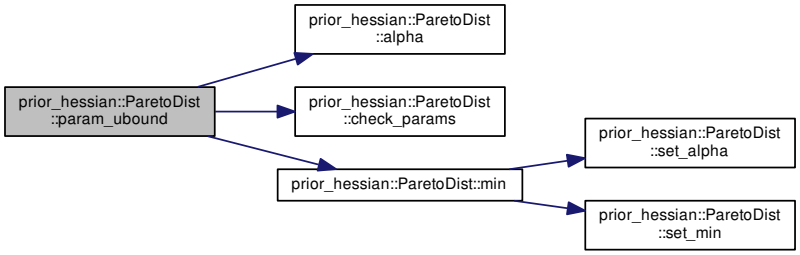
Definition at line 31 of file ParetoDist.h.

7.56.4.26 static const NparamsVecT& prior_hessian::ParetoDist::param_ubound ( ) [inline], [static]

Definition at line 33 of file ParetoDist.h.

References alpha(), check_params(), and min().

Here is the call graph for this function:



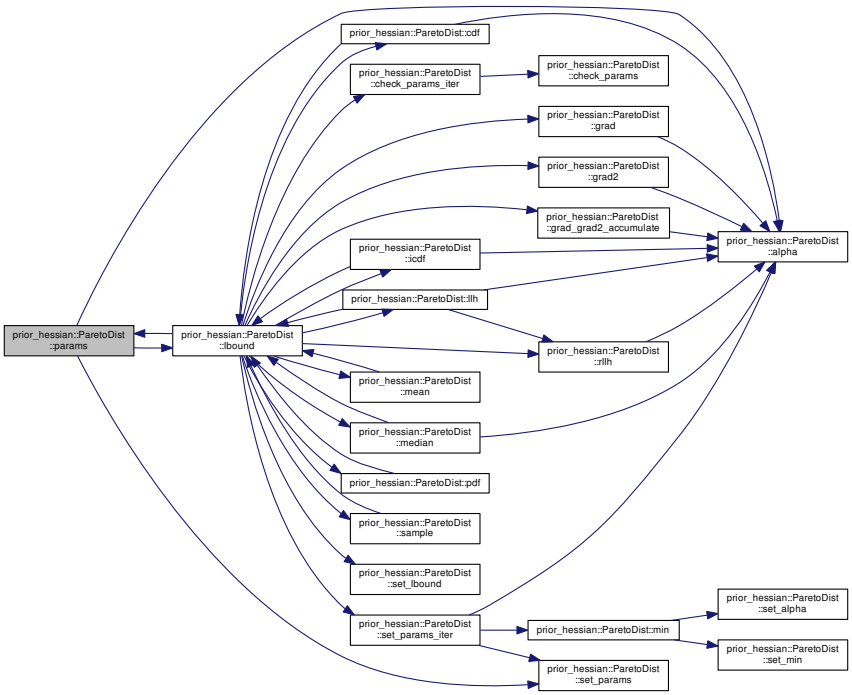
7.56.4.27 NparamsVecT prior_hessian::ParetoDist::params ( ) const [inline]

Definition at line 48 of file ParetoDist.h.

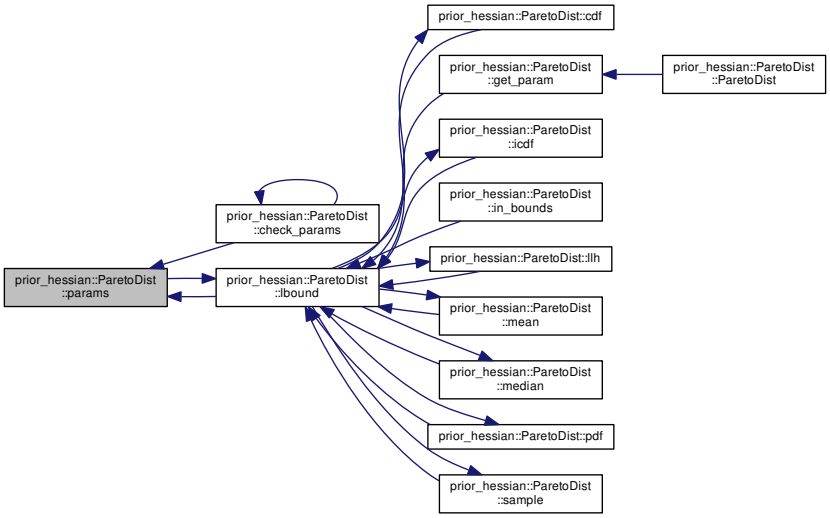
References alpha(), lbound(), and set_params().

Referenced by check_params(), and lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



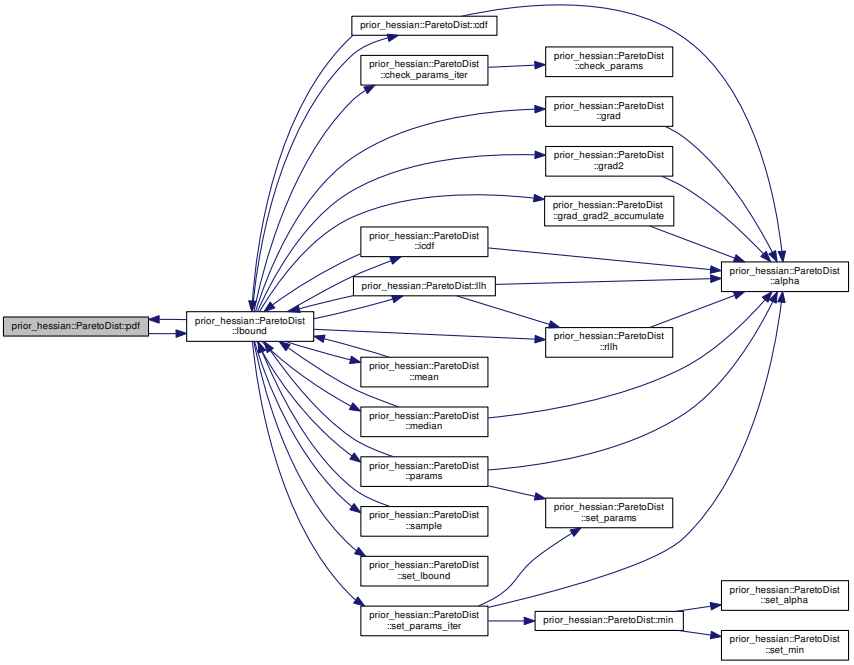
7.56.4.28 double prior_hessian::ParetoDist::pdf (double *x*) const [inline]

Definition at line 167 of file ParetoDist.h.

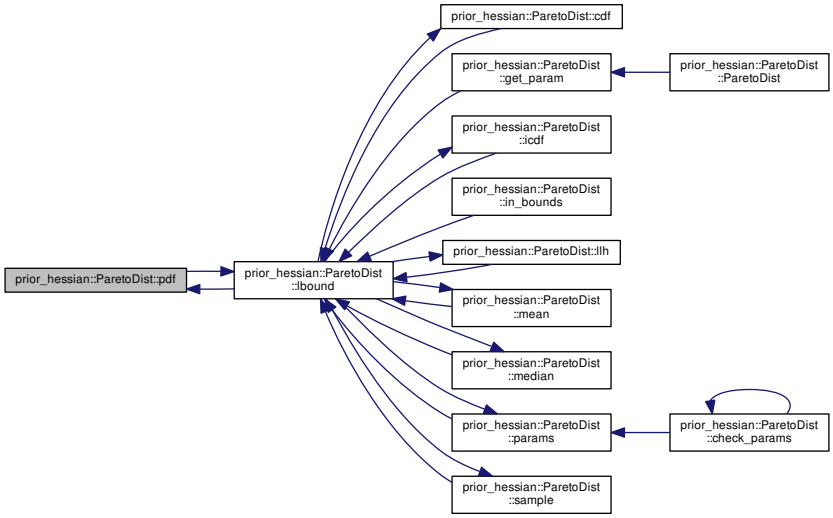
References lbound().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



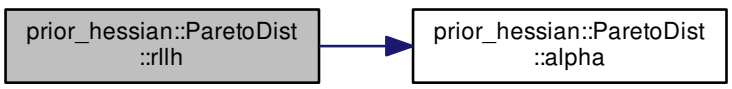
7.56.4.29 double prior_hessian::ParetoDist::rllh (double *x*) const [inline]

Definition at line 173 of file ParetoDist.h.

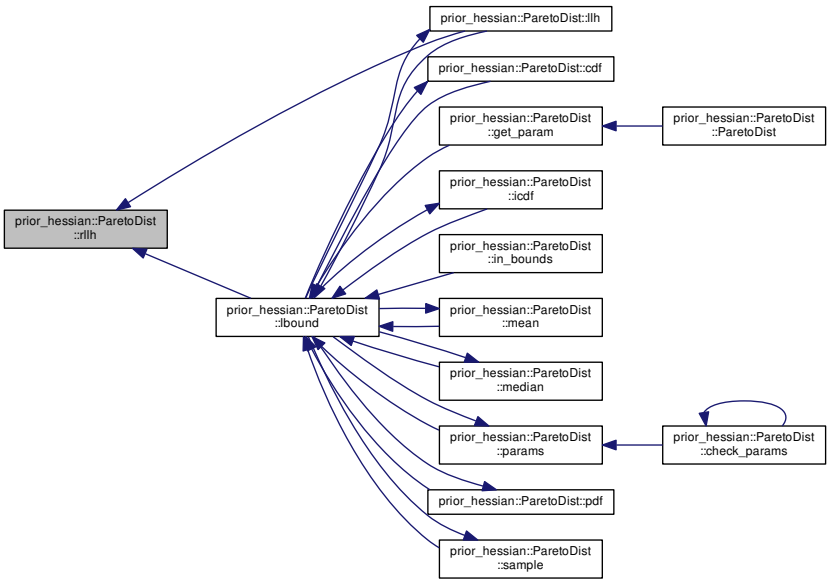
References alpha().

Referenced by lbound(), and llh().

Here is the call graph for this function:



Here is the caller graph for this function:



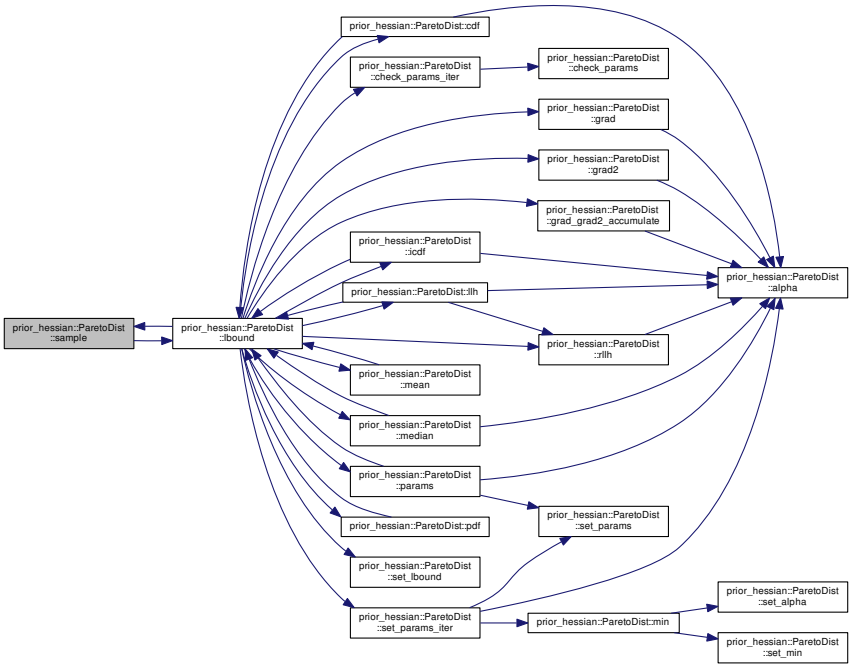
7.56.4.30 template<class RngT > double prior_hessian::ParetoDist::sample (RngT & *rng*) const

Definition at line 200 of file ParetoDist.h.

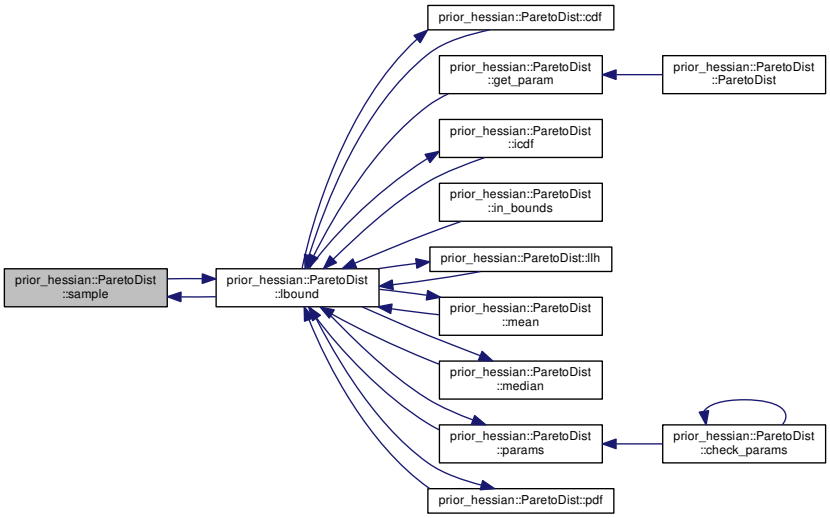
References lbound().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:

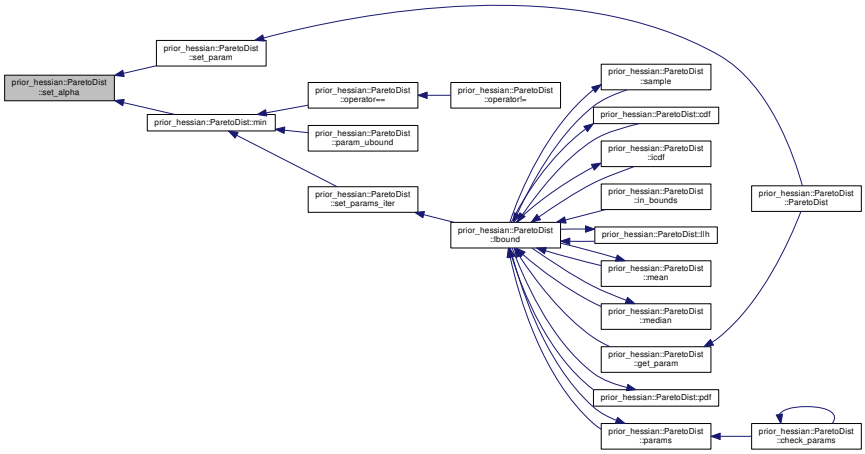


7.56.4.31 void prior_hessian::ParetoDist::set_alpha (double *val*)

Definition at line 35 of file ParetoDist.cpp.

Referenced by min(), and set_param().

Here is the caller graph for this function:

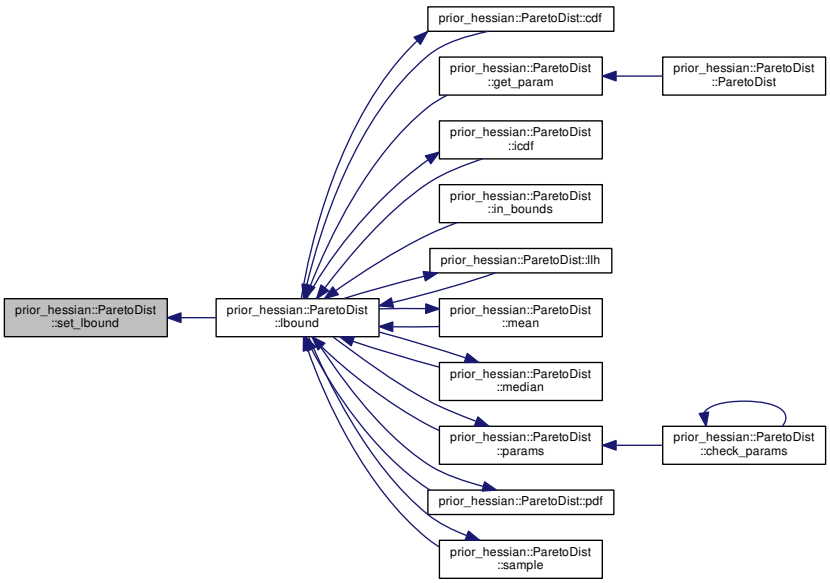


7.56.4.32 void prior_hessian::ParetoDist::set_lbound (double *lbound*)

Definition at line 48 of file ParetoDist.cpp.

Referenced by lbound().

Here is the caller graph for this function:

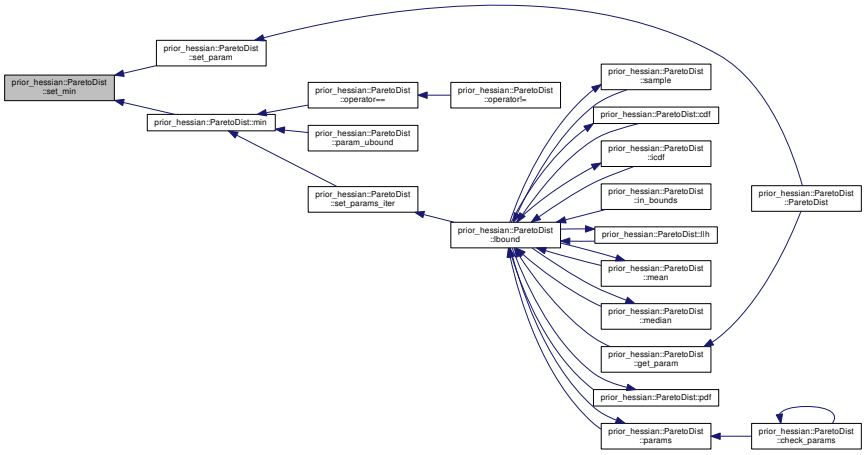


7.56.4.33 void prior_hessian::ParetoDist::set_min (double *val*)

Definition at line 29 of file ParetoDist.cpp.

Referenced by min(), and set_param().

Here is the caller graph for this function:



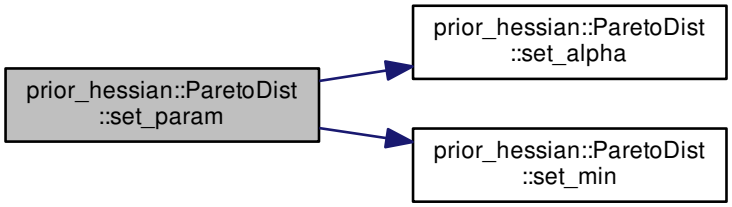
7.56.4.34 void prior_hessian::ParetoDist::set_param (int *idx*, double *val*) [inline]

Definition at line 133 of file ParetoDist.h.

References set_alpha(), and set_min().

Referenced by ParetoDist().

Here is the call graph for this function:



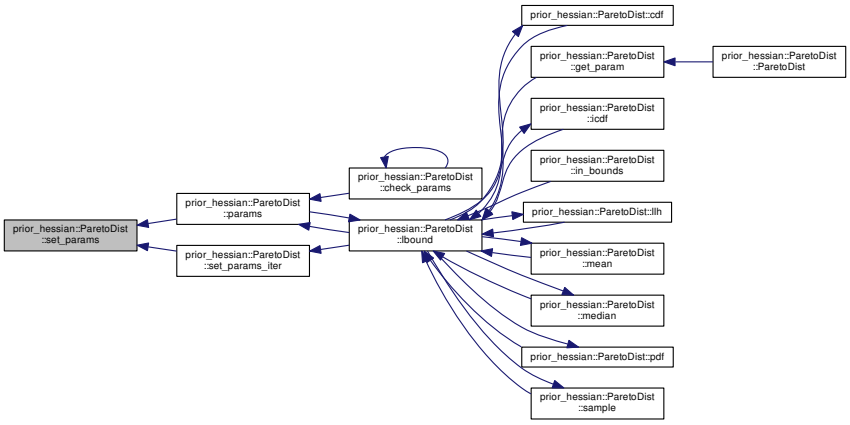
Here is the caller graph for this function:

7.56.4.35 void prior_hessian::ParetoDist::set_params (double *min,* double *alpha*)

Definition at line 41 of file ParetoDist.cpp.

Referenced by params(), and set_params_iter().

Here is the caller graph for this function:



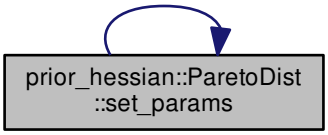
7.56.4.36 template<class Vec > void prior_hessian::ParetoDist::set_params (const Vec & *p*) [inline]

Definition at line 51 of file ParetoDist.h.

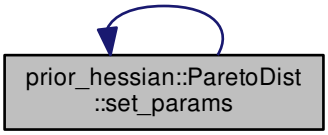
References set_params().

Referenced by set_params().

Here is the call graph for this function:



Here is the caller graph for this function:



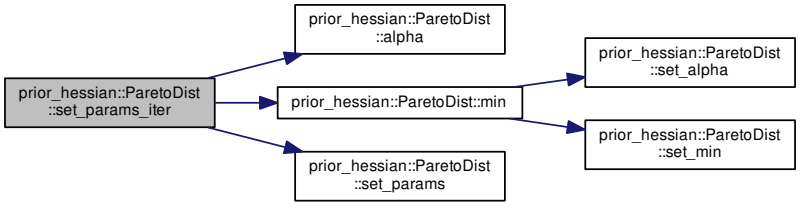
7.56.4.37 template<class IterT > void prior_hessian::ParetoDist::set_params_iter (IterT & *params*)

Definition at line 208 of file ParetoDist.h.

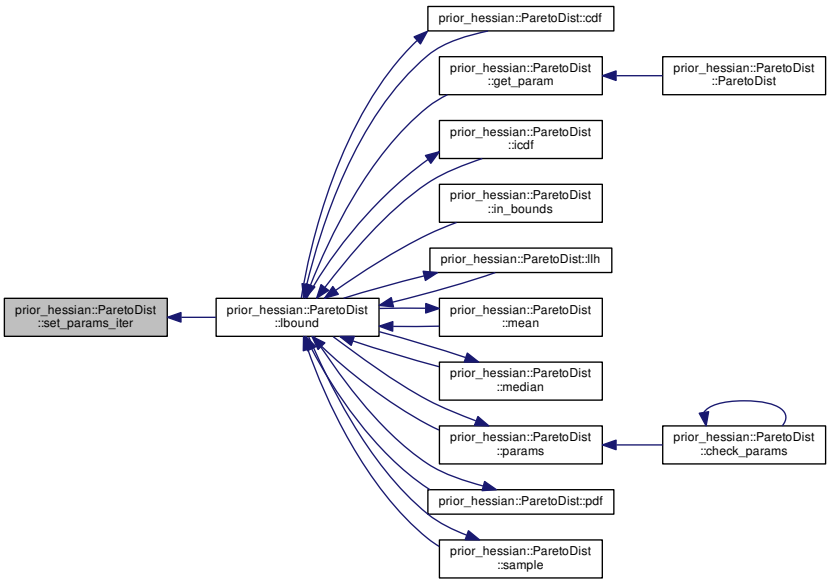
References alpha(), min(), and set_params().

Referenced by lbound().

Here is the call graph for this function:



Here is the caller graph for this function:



7.56.4.38 static constexpr double prior_hessian::ParetoDist::ubound ( ) [inline], [static]

Definition at line 27 of file ParetoDist.h.

Referenced by in_bounds().

Here is the caller graph for this function:

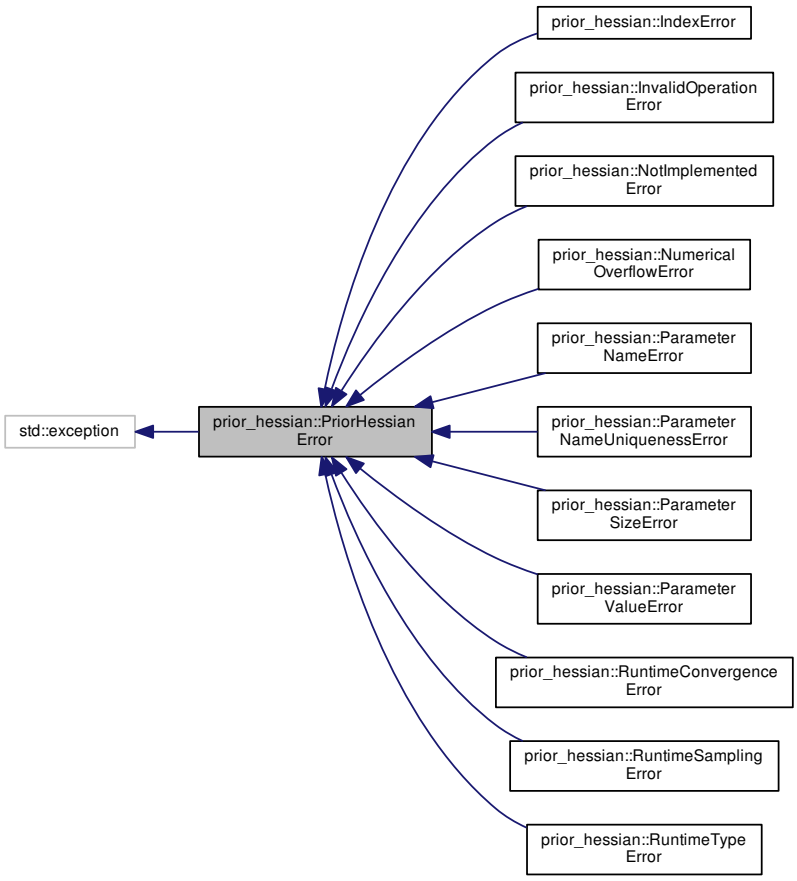
The documentation for this class was generated from the following files:

- ParetoDist.h
- ParetoDist.cpp

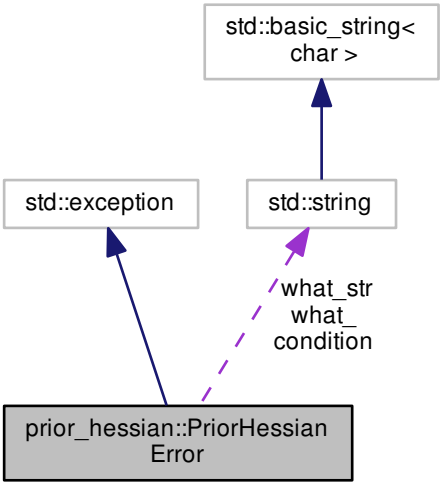
7.57 prior_hessian::PriorHessianError Class Reference

```
#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩
PriorHessianError.h>
```

Inheritance diagram for prior_hessian::PriorHessianError:



Collaboration diagram for prior_hessian::PriorHessianError:



Public Member Functions

- PriorHessianError (std::string condition, std::string what)
- const char ∗ what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.57.1 Detailed Description

Definition at line 43 of file PriorHessianError.h.

7.57.2 Constructor & Destructor Documentation

7.57.2.1 prior_hessian::PriorHessianError::PriorHessianError (std::string *condition,* std::string *what*) [inline]

Definition at line 50 of file PriorHessianError.h.

7.57.3 Member Function Documentation

7.57.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept]

Definition at line 56 of file PriorHessianError.h.

7.57.4 Member Data Documentation

7.57.4.1 std::string prior_hessian::PriorHessianError::condition [protected]

Definition at line 46 of file PriorHessianError.h.

7.57.4.2 std::string prior_hessian::PriorHessianError::what_ [protected]

Definition at line 48 of file PriorHessianError.h.

7.57.4.3 std::string prior_hessian::PriorHessianError::what_str [protected]

Definition at line 47 of file PriorHessianError.h.

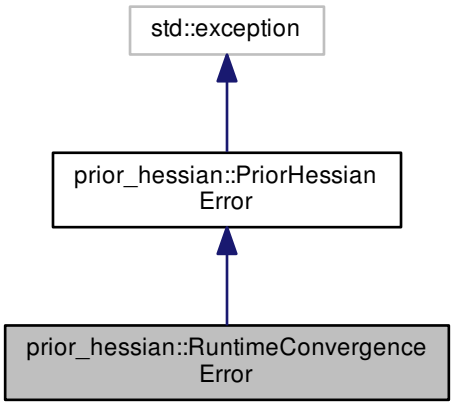
The documentation for this class was generated from the following file:

- PriorHessianError.h

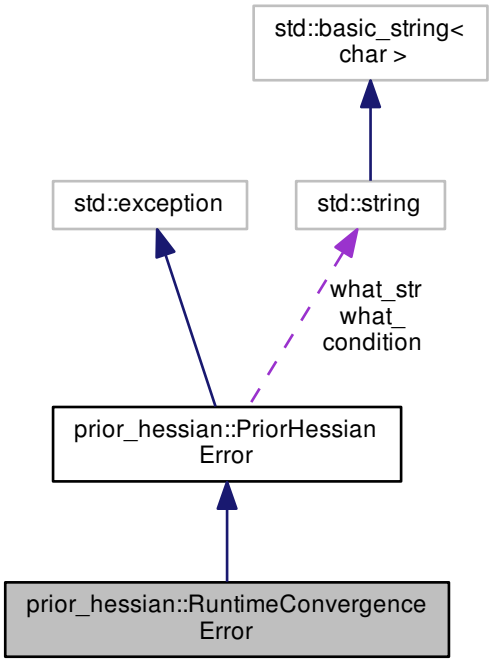
7.58 prior_hessian::RuntimeConvergenceError Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::RuntimeConvergenceError:



Collaboration diagram for prior_hessian::RuntimeConvergenceError:



Public Member Functions

- RuntimeConvergenceError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.58.1 Detailed Description

Definition at line 94 of file PriorHessianError.h.

7.58.2 Constructor & Destructor Documentation

7.58.2.1 prior_hessian::RuntimeConvergenceError::RuntimeConvergenceError (std::string *message*) [inline]

Definition at line 96 of file PriorHessianError.h.

7.58.3 Member Function Documentation

7.58.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.58.4 Member Data Documentation

7.58.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.58.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.58.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

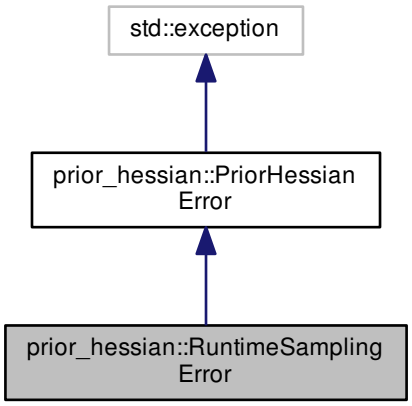
The documentation for this struct was generated from the following file:

- PriorHessianError.h

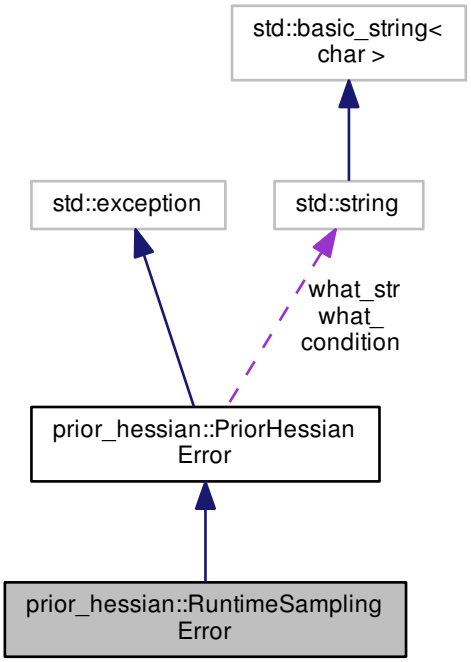
7.59 prior_hessian::RuntimeSamplingError Struct Reference

```
#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩
PriorHessianError.h>
```

Inheritance diagram for prior_hessian::RuntimeSamplingError:



Collaboration diagram for prior_hessian::RuntimeSamplingError:



Public Member Functions

- RuntimeSamplingError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.59.1 Detailed Description

Definition at line 99 of file PriorHessianError.h.

7.59.2 Constructor & Destructor Documentation

7.59.2.1 prior_hessian::RuntimeSamplingError::RuntimeSamplingError (std::string *message*) [inline]

Definition at line 101 of file PriorHessianError.h.

7.59.3 Member Function Documentation

7.59.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.59.4 Member Data Documentation

7.59.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.59.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.59.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

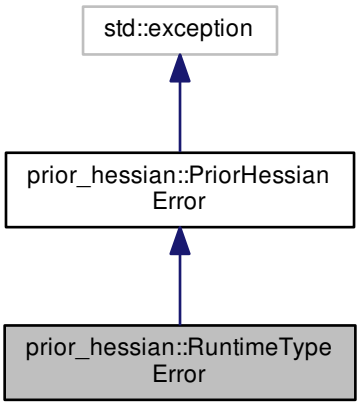
The documentation for this struct was generated from the following file:

- PriorHessianError.h

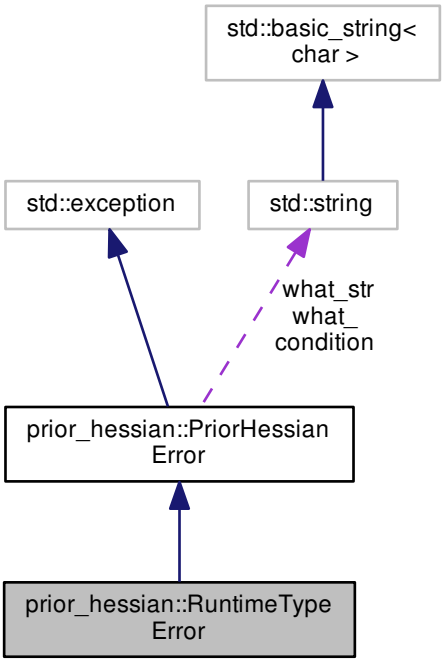
7.60 prior_hessian::RuntimeTypeError Struct Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ PriorHessianError.h>`

Inheritance diagram for prior_hessian::RuntimeTypeError:



Collaboration diagram for prior_hessian::RuntimeTypeError:



Public Member Functions

- RuntimeTypeError (std::string message)
- const char * what () const noexceptoverride

Protected Attributes

- std::string condition
- std::string what_str
- std::string what_

7.60.1 Detailed Description

Definition at line 104 of file PriorHessianError.h.

7.60.2 Constructor & Destructor Documentation

7.60.2.1 prior_hessian::RuntimeTypeError::RuntimeTypeError (std::string *message*) [inline]

Definition at line 106 of file PriorHessianError.h.

7.60.3 Member Function Documentation

7.60.3.1 const char* prior_hessian::PriorHessianError::what ( ) const [inline], [override], [noexcept], [inherited]

Definition at line 56 of file PriorHessianError.h.

7.60.4 Member Data Documentation

7.60.4.1 std::string prior_hessian::PriorHessianError::condition [protected], [inherited]

Definition at line 46 of file PriorHessianError.h.

7.60.4.2 std::string prior_hessian::PriorHessianError::what_ [protected], [inherited]

Definition at line 48 of file PriorHessianError.h.

7.60.4.3 std::string prior_hessian::PriorHessianError::what_str [protected], [inherited]

Definition at line 47 of file PriorHessianError.h.

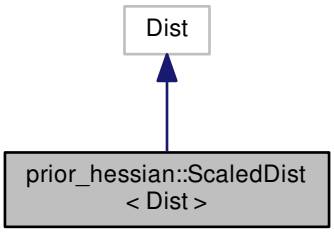
The documentation for this struct was generated from the following file:

- PriorHessianError.h

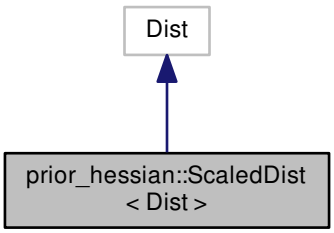
7.61 prior_hessian::ScaledDist< Dist > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/ScaledDist.h>`

Inheritance diagram for prior_hessian::ScaledDist< Dist >:



Collaboration diagram for prior_hessian::ScaledDist< Dist >:



Protected Member Functions

- double convert_to_unitary_coords (double x) const
- double convert_from_unitary_coords (double u) const

Protected Attributes

- double _scaled_lbound
- double _scaled_ubound
- double scaling_ratio
- double llh_scaling_const

7.61.1 Detailed Description

template<class Dist>
class prior_hessian::ScaledDist< Dist >

Definition at line 19 of file ScaledDist.h.

7.61.2 Constructor & Destructor Documentation

7.61.2.1 template<class Dist> prior_hessian::ScaledDist< Dist >::ScaledDist ( ) [inline]

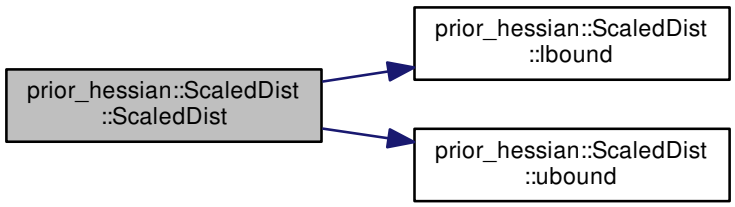
Definition at line 22 of file ScaledDist.h.

7.61.2.2 template<class Dist> prior_hessian::ScaledDist< Dist >::ScaledDist (double *lbound*, double *ubound*) [inline]

Definition at line 23 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::lbound(), and prior_hessian::ScaledDist< Dist >::ubound().

Here is the call graph for this function:



7.61.2.3 template<class Dist> template<typename = meta::EnableIfNotIsSelfT<Dist,ScaledDist>> prior_hessian::ScaledDist< Dist >::ScaledDist (const Dist & *dist*) [inline]

Definition at line 26 of file ScaledDist.h.

7.61.2.4 template<class Dist> template<typename = meta::EnableIfNotIsSelfT<Dist,ScaledDist>> prior_hessian::ScaledDist< Dist >::ScaledDist (Dist && *dist*) [inline]

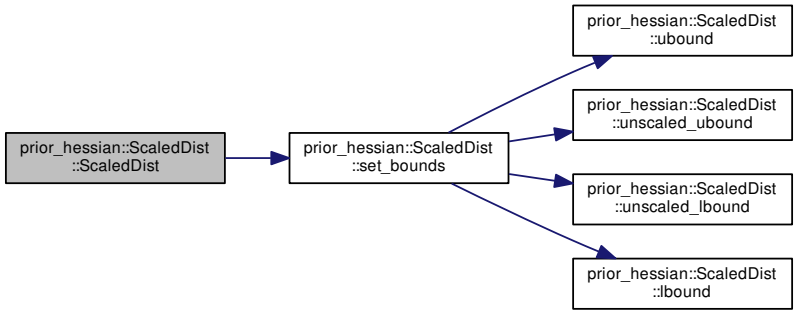
Definition at line 29 of file ScaledDist.h.

7.61.2.5 template<class Dist> prior_hessian::ScaledDist< Dist >::ScaledDist (const Dist & *dist,* double *lbound,* double *ubound*) [inline]

Definition at line 31 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::set_bounds().

Here is the call graph for this function:

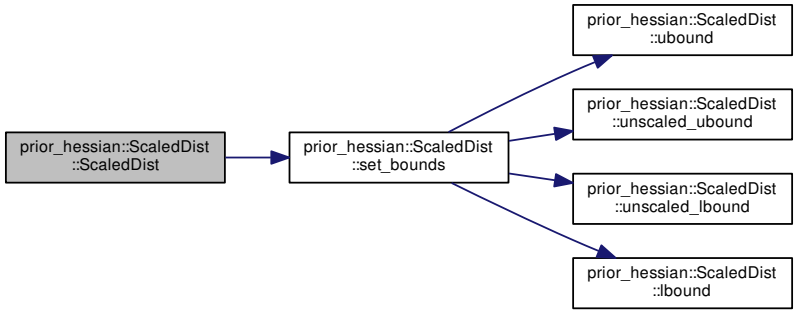


7.61.2.6 template<class Dist> prior_hessian::ScaledDist< Dist >::ScaledDist (Dist && *dist,* double *lbound,* double *ubound*) [inline]

Definition at line 32 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::set_bounds().

Here is the call graph for this function:



7.61.3 Member Function Documentation

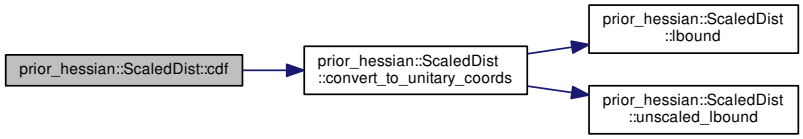
7.61.3.1 template<class Dist > double prior_hessian::ScaledDist< Dist >::cdf (double *x*) const

Definition at line 112 of file ScaledDist.h.

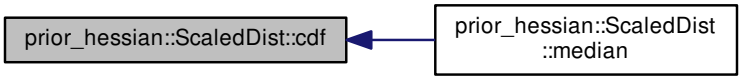
References prior_hessian::ScaledDist< Dist >::convert_to_unitary_coords().

Referenced by prior_hessian::ScaledDist< Dist >::median().

Here is the call graph for this function:



Here is the caller graph for this function:



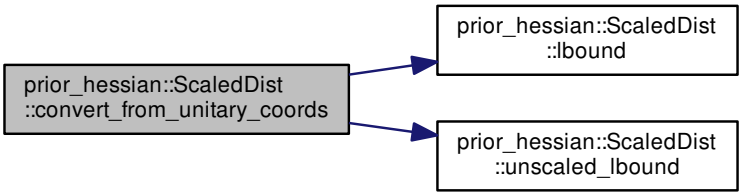
7.61.3.2 template<class Dist > double prior_hessian::ScaledDist< Dist >::convert_from_unitary_coords (double *u*) const [protected]

Definition at line 143 of file ScaledDist.h.

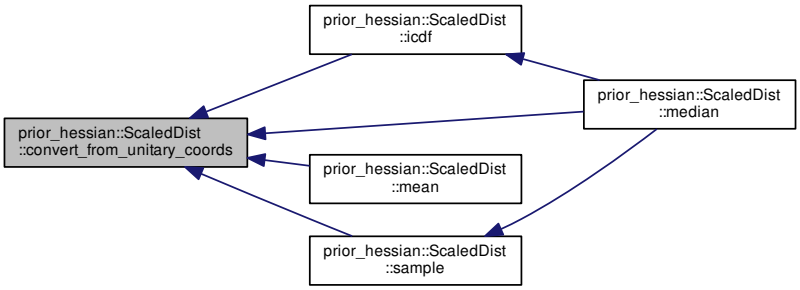
References prior_hessian::ScaledDist< Dist >::lbound(), prior_hessian::ScaledDist< Dist >::scaling_ratio, and prior_hessian::ScaledDist< Dist >::unscaled_lbound().

Referenced by prior_hessian::ScaledDist< Dist >::icdf(), prior_hessian::ScaledDist< Dist >::mean(), prior_hessian::ScaledDist< Dist >::median(), and prior_hessian::ScaledDist< Dist >::sample().

Here is the call graph for this function:



Here is the caller graph for this function:



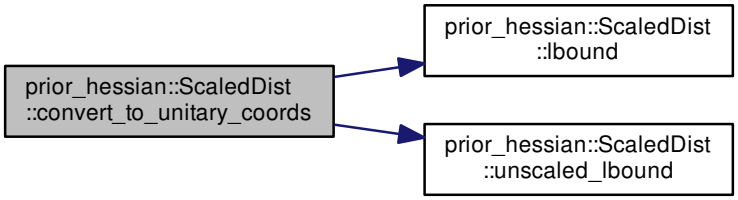
7.61.3.3 template<class Dist > double prior_hessian::ScaledDist< Dist >::convert_to_unitary_coords (double *x*) const [protected]

Definition at line 137 of file ScaledDist.h.

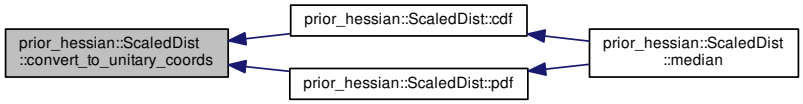
References prior_hessian::ScaledDist< Dist >::lbound(), prior_hessian::ScaledDist< Dist >::scaling_ratio, and prior_hessian::ScaledDist< Dist >::unscaled_lbound().

Referenced by prior_hessian::ScaledDist< Dist >::cdf(), and prior_hessian::ScaledDist< Dist >::pdf().

Here is the call graph for this function:



Here is the caller graph for this function:



7.61.3.4 template<class Dist> static double prior_hessian::ScaledDist< Dist >::global_lbound ( ) [inline], [static]

Definition at line 38 of file ScaledDist.h.

7.61.3.5 template<class Dist> static double prior_hessian::ScaledDist< Dist >::global_ubound ( ) [inline], [static]

Definition at line 39 of file ScaledDist.h.

7.61.3.6 template<class Dist > double prior_hessian::ScaledDist< Dist >::icdf (double *u*) const

Definition at line 118 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::convert_from_unitary_coords().

Referenced by prior_hessian::ScaledDist< Dist >::median().

Here is the call graph for this function:

Here is the caller graph for this function:

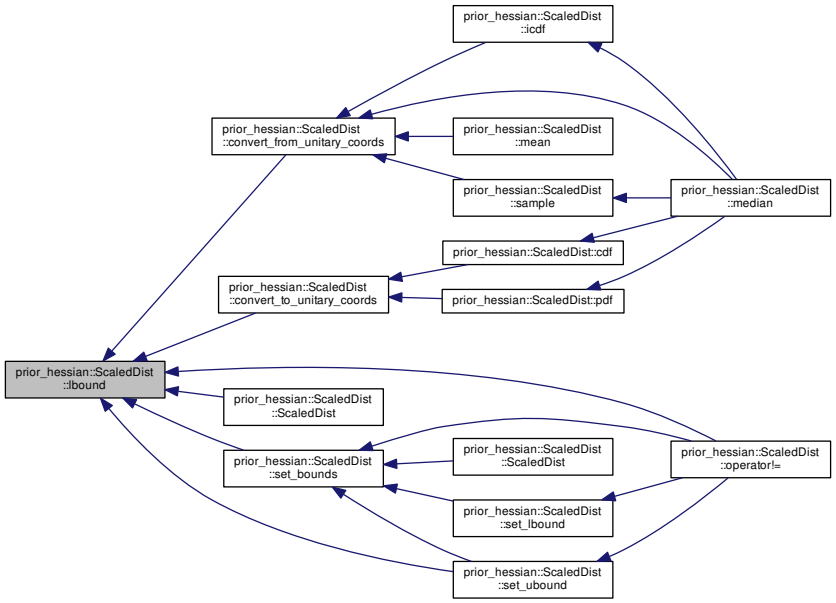
7.61.3.7 template<class Dist> double prior_hessian::ScaledDist< Dist >::lbound ( ) const [inline]

Definition at line 34 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::_scaled_lbound.

Referenced by prior_hessian::ScaledDist< Dist >::convert_from_unitary_coords(), prior_hessian::ScaledDist< Dist >::convert_to_unitary_coords(), prior_hessian::ScaledDist< Dist >::operator!=(), prior_hessian::ScaledDist< Dist >::ScaledDist(), prior_hessian::ScaledDist< Dist >::set_bounds(), and prior_hessian::ScaledDist< Dist >::set_↩ ubound().

Here is the caller graph for this function:



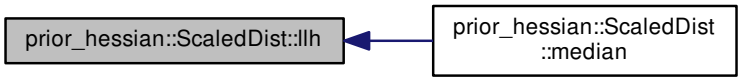
7.61.3.8 template<class Dist > double prior_hessian::ScaledDist< Dist >::llh (double *x*) const

Definition at line 106 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::llh_scaling_const.

Referenced by prior_hessian::ScaledDist< Dist >::median().

Here is the caller graph for this function:

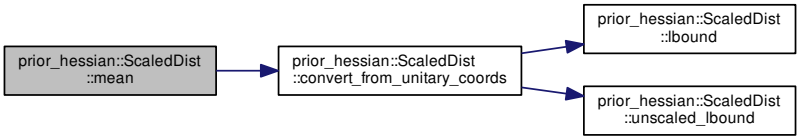


7.61.3.9 template<class Dist> double prior_hessian::ScaledDist< Dist >::mean ( ) const [inline]

Definition at line 53 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::convert_from_unitary_coords().

Here is the call graph for this function:

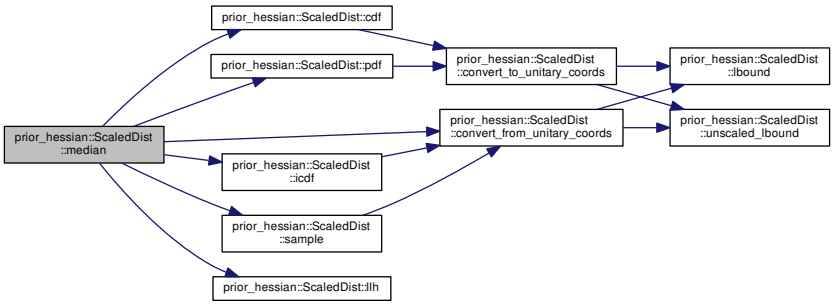


7.61.3.10 template<class Dist> double prior_hessian::ScaledDist< Dist >::median ( ) const [inline]

Definition at line 54 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::cdf(), prior_hessian::ScaledDist< Dist >::convert_from_unitary_↩ coords(), prior_hessian::ScaledDist< Dist >::icdf(), prior_hessian::ScaledDist< Dist >::llh(), prior_hessian::Scaled↩ Dist< Dist >::pdf(), and prior_hessian::ScaledDist< Dist >::sample().

Here is the call graph for this function:

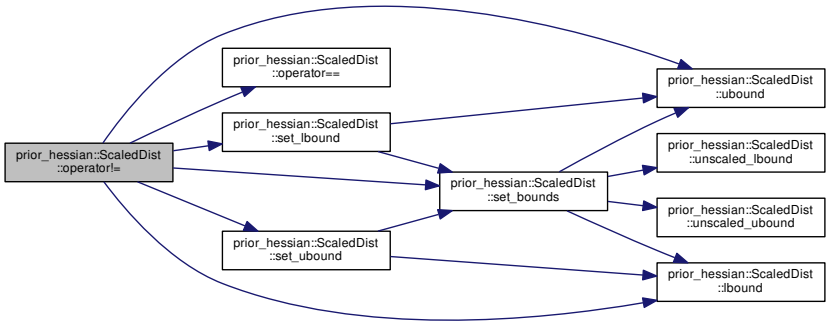


7.61.3.11 template<class Dist> bool prior_hessian::ScaledDist< Dist >::operator!= (const ScaledDist< Dist > & *o*) const [inline]

Definition at line 47 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::lbound(), prior_hessian::ScaledDist< Dist >::operator==(), prior_↩ hessian::ScaledDist< Dist >::set_bounds(), prior_hessian::ScaledDist< Dist >::set_lbound(), prior_hessian::Scaled↩ Dist< Dist >::set_ubound(), and prior_hessian::ScaledDist< Dist >::ubound().

Here is the call graph for this function:



7.61.3.12 template<class Dist> bool prior_hessian::ScaledDist< Dist >::operator== (const ScaledDist< Dist > & *o*) const [inline]

Definition at line 41 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::_scaled_lbound, and prior_hessian::ScaledDist< Dist >::_scaled_↩ ubound.

Referenced by prior_hessian::ScaledDist< Dist >::operator!=().

Here is the caller graph for this function:

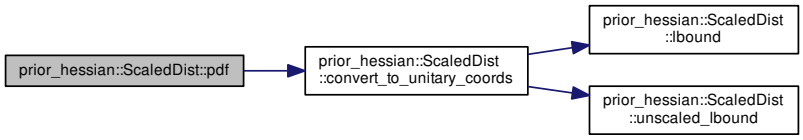
7.61.3.13 template<class Dist > double prior_hessian::ScaledDist< Dist >::pdf (double *x*) const

Definition at line 124 of file ScaledDist.h.

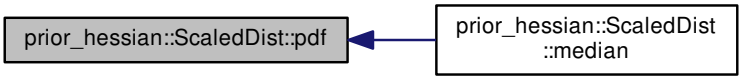
References prior_hessian::ScaledDist< Dist >::convert_to_unitary_coords(), and prior_hessian::ScaledDist< Dist >↩ ::scaling_ratio.

Referenced by prior_hessian::ScaledDist< Dist >::median().

Here is the call graph for this function:



Here is the caller graph for this function:



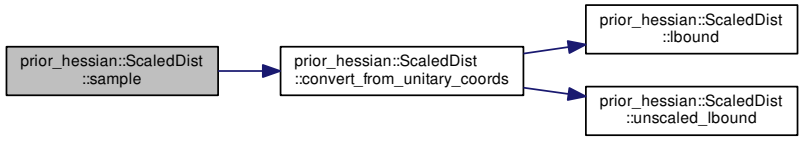
7.61.3.14 template<class Dist > template<class RngT > double prior_hessian::ScaledDist< Dist >::sample (RngT & *rng*) const

Definition at line 131 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::convert_from_unitary_coords().

Referenced by prior_hessian::ScaledDist< Dist >::median().

Here is the call graph for this function:



Here is the caller graph for this function:

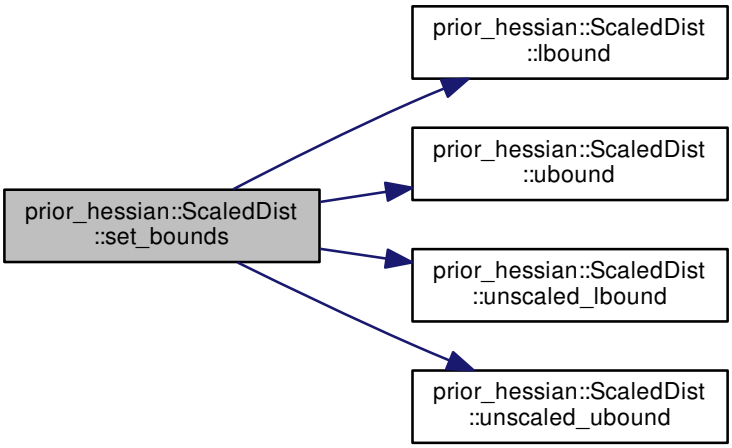
7.61.3.15 template<class Dist > void prior_hessian::ScaledDist< Dist >::set_bounds (double *lbound,* double *ubound*)

Definition at line 76 of file ScaledDist.h.

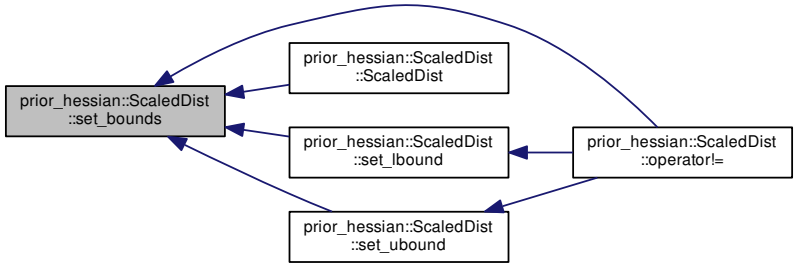
References prior_hessian::ScaledDist< Dist >::_scaled_lbound, prior_hessian::ScaledDist< Dist >::_scaled_ubound, prior_hessian::ScaledDist< Dist >::lbound(), prior_hessian::ScaledDist< Dist >::llh_scaling_const, prior_hessian::ScaledDist< Dist >::scaling_ratio, prior_hessian::ScaledDist< Dist >::ubound(), prior_hessian::ScaledDist< Dist >::unscaled_lbound(), and prior_hessian::ScaledDist< Dist >::unscaled_ubound().

Referenced by prior_hessian::ScaledDist< Dist >::operator!=(), prior_hessian::ScaledDist< Dist >::ScaledDist(), prior_hessian::ScaledDist< Dist >::set_lbound(), and prior_hessian::ScaledDist< Dist >::set_ubound().

Here is the call graph for this function:



Here is the caller graph for this function:



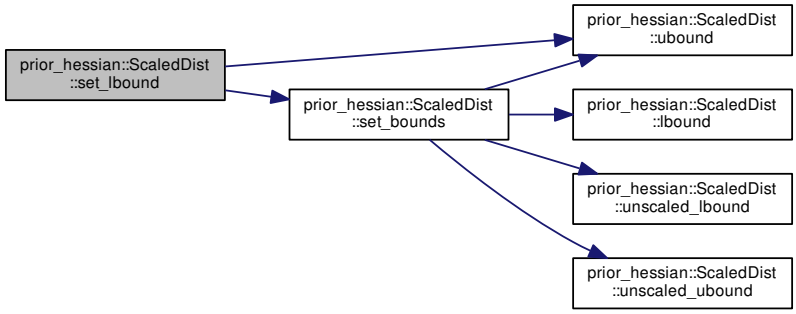
7.61.3.16 template<class Dist > void prior_hessian::ScaledDist< Dist >::set_lbound (double *lbound*)

Definition at line 98 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::set_bounds(), and prior_hessian::ScaledDist< Dist >::ubound().

Referenced by prior_hessian::ScaledDist< Dist >::operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

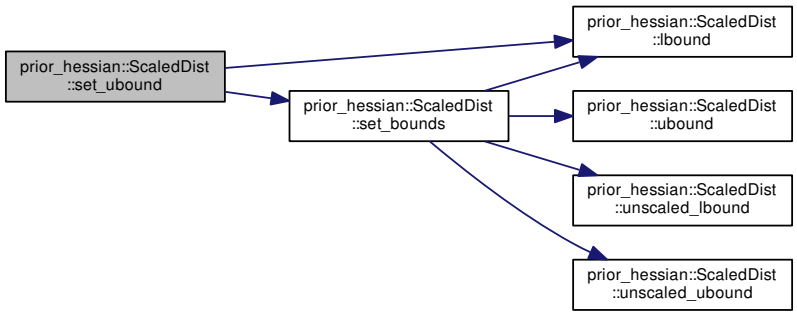
7.61.3.17 template<class Dist > void prior_hessian::ScaledDist< Dist >::set_ubound (double *ubound*)

Definition at line 102 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::lbound(), and prior_hessian::ScaledDist< Dist >::set_bounds().

Referenced by prior_hessian::ScaledDist< Dist >::operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

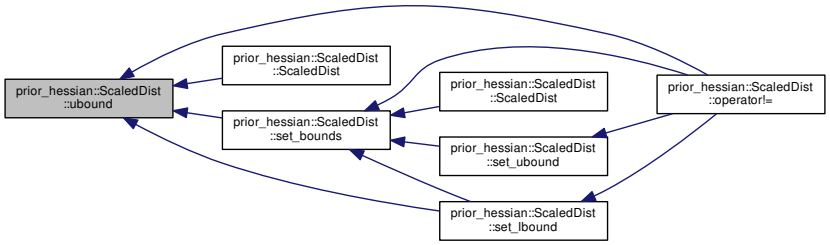
7.61.3.18 template<class Dist> double prior_hessian::ScaledDist< Dist >::ubound ( ) const [inline]

Definition at line 35 of file ScaledDist.h.

References prior_hessian::ScaledDist< Dist >::_scaled_ubound.

Referenced by prior_hessian::ScaledDist< Dist >::operator!=(), prior_hessian::ScaledDist< Dist >::ScaledDist(), prior_hessian::ScaledDist< Dist >::set_bounds(), and prior_hessian::ScaledDist< Dist >::set_lbound().

Here is the caller graph for this function:

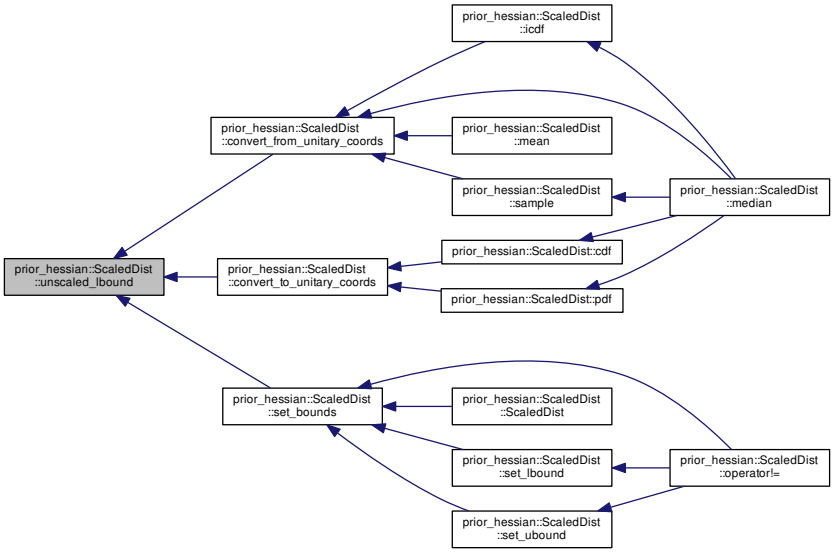


7.61.3.19 template<class Dist> static double prior_hessian::ScaledDist< Dist >::unscaled_lbound ( ) [inline], [static]

Definition at line 36 of file ScaledDist.h.

Referenced by prior_hessian::ScaledDist< Dist >::convert_from_unitary_coords(), prior_hessian::ScaledDist< Dist >::convert_to_unitary_coords(), and prior_hessian::ScaledDist< Dist >::set_bounds().

Here is the caller graph for this function:

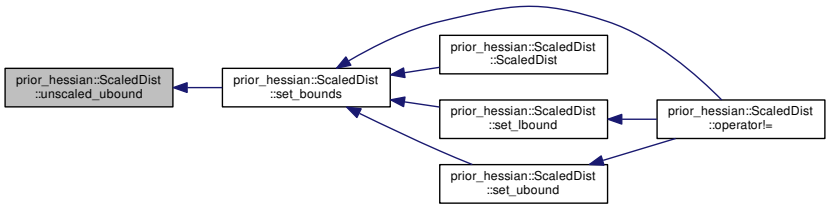


7.61.3.20 template<class Dist> static double prior_hessian::ScaledDist< Dist >::unscaled_ubound ( ) [inline], [static]

Definition at line 37 of file ScaledDist.h.

Referenced by prior_hessian::ScaledDist< Dist >::set_bounds().

Here is the caller graph for this function:



7.61.4 Member Data Documentation

7.61.4.1 template<class Dist> double prior_hessian::ScaledDist< Dist >::_scaled_lbound [protected]

Definition at line 64 of file ScaledDist.h.

Referenced by prior_hessian::ScaledDist< Dist >::lbound(), prior_hessian::ScaledDist< Dist >::operator==(), and prior_hessian::ScaledDist< Dist >::set_bounds().

7.61.4.2 template<class Dist> double prior_hessian::ScaledDist< Dist >::_scaled_ubound [protected]

Definition at line 65 of file ScaledDist.h.

Referenced by prior_hessian::ScaledDist< Dist >::operator==(), prior_hessian::ScaledDist< Dist >::set_bounds(), and prior_hessian::ScaledDist< Dist >::ubound().

7.61.4.3 template<class Dist> double prior_hessian::ScaledDist< Dist >::llh_scaling_const [protected]

Definition at line 68 of file ScaledDist.h.

Referenced by prior_hessian::ScaledDist< Dist >::llh(), and prior_hessian::ScaledDist< Dist >::set_bounds().

7.61.4.4 template<class Dist> double prior_hessian::ScaledDist< Dist >::scaling_ratio [protected]

Definition at line 67 of file ScaledDist.h.

Referenced by prior_hessian::ScaledDist< Dist >::convert_from_unitary_coords(), prior_hessian::ScaledDist< Dist >::convert_to_unitary_coords(), prior_hessian::ScaledDist< Dist >::pdf(), and prior_hessian::ScaledDist< Dist >::set_bounds().

The documentation for this class was generated from the following file:

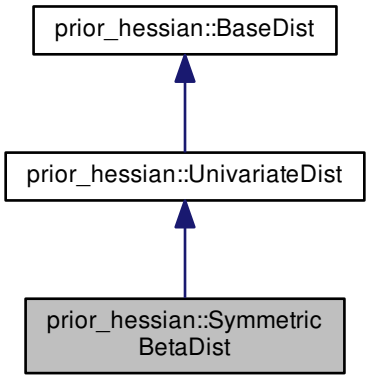
- ScaledDist.h

7.62 prior_hessian::SymmetricBetaDist Class Reference

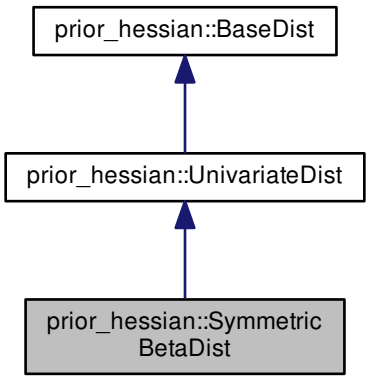
Single parameter beta distribution where $\alpha = \beta$, leading to symmetric bounded distribution.

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/SymmetricBetaDist.h>`

Inheritance diagram for prior_hessian::SymmetricBetaDist:



Collaboration diagram for prior_hessian::SymmetricBetaDist:



Public Types

- using NparamsVecT = arma::Col< double >::fixed< _num_params >

Public Member Functions

- SymmetricBetaDist ()
- SymmetricBetaDist (double beta)
- template<class Vec , meta::ConstructableIfNotSelfT< Vec, SymmetricBetaDist > = true>
  SymmetricBetaDist (const Vec &params)
- double beta () const
- void set_beta (double val)
- double get_param (int idx) const
- void set_param (int idx, double val)
- NparamsVecT params () const
- void set_params (double beta)
- template<class Vec >
  void set_params (const Vec &p)
- bool operator== (const SymmetricBetaDist &o) const
- bool operator!= (const SymmetricBetaDist &o) const
- double mean () const
- double median () const
- double cdf (double x) const
- double icdf (double u) const
- double pdf (double x) const
- double llh (double x) const
- double rllh (double x) const
- double grad (double x) const
- double grad2 (double x) const
- void grad_grad2_accumulate (double x, double &g, double &g2) const
- template<class RngT >
  double sample (RngT &rng) const
- template<class IterT >
  void set_params_iter (IterT &params)

Static Public Member Functions

- static constexpr IdxT num_params ()
- static constexpr double lbound ()
- static constexpr double ubound ()
- static bool in_bounds (double u)
- static const StringVecT & param_names ()
- static const VecT & param_lbound ()
- static const VecT & param_ubound ()
- static bool check_params (double beta)
- template<class Vec >
  static bool check_params (Vec &p)
- template<class IterT >
  static bool check_params_iter (IterT &params)
- static constexpr IdxT num_dim ()

Static Protected Member Functions

- static void check_bounds (double lbound, double ubound)

7.62.1 Detailed Description

Single parameter beta distribution where $\alpha = \beta$, leading to symmetric bounded distribution.

Definition at line 22 of file SymmetricBetaDist.h.

7.62.2 Member Typedef Documentation

7.62.2.1 using prior_hessian::SymmetricBetaDist::NparamsVecT = arma::Col<double>::fixed<_num_params>

Definition at line 26 of file SymmetricBetaDist.h.

7.62.3 Constructor & Destructor Documentation

7.62.3.1 prior_hessian::SymmetricBetaDist::SymmetricBetaDist ( ) [inline]

Definition at line 42 of file SymmetricBetaDist.h.

7.62.3.2 prior_hessian::SymmetricBetaDist::SymmetricBetaDist (double *beta*) [explicit]

Definition at line 23 of file SymmetricBetaDist.cpp.

7.62.3.3 template<class Vec , meta::ConstructableIfNotSelfT< Vec, SymmetricBetaDist > = true> prior_hessian::SymmetricBetaDist::SymmetricBetaDist (const Vec & *params*) [inline], [explicit]

Definition at line 45 of file SymmetricBetaDist.h.

7.62.4 Member Function Documentation

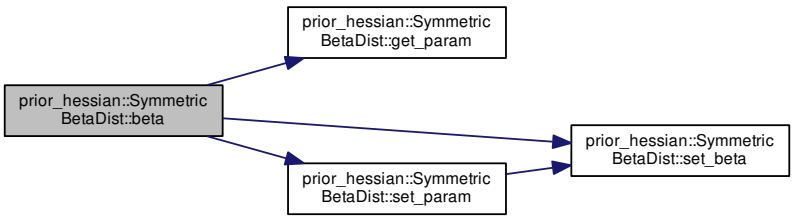
7.62.4.1 double prior_hessian::SymmetricBetaDist::beta ( ) const [inline]

Definition at line 47 of file SymmetricBetaDist.h.

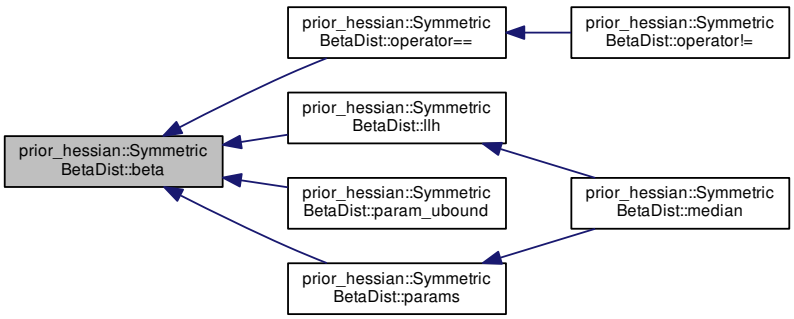
References get_param(), set_beta(), and set_param().

Referenced by llh(), operator==(), param_ubound(), and params().

Here is the call graph for this function:



Here is the caller graph for this function:



7.62.4.2 double prior_hessian::SymmetricBetaDist::cdf (double *x*) const

Definition at line 36 of file SymmetricBetaDist.cpp.

Referenced by median().

Here is the caller graph for this function:

7.62.4.3 void prior_hessian::UnivariateDist::check_bounds (double *lbound,* double *ubound*) [static],[protected], [inherited]

Definition at line 17 of file UnivariateDist.cpp.

Referenced by prior_hessian::UnivariateDist::UnivariateDist().

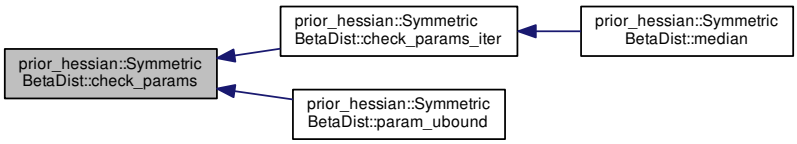
Here is the caller graph for this function:

7.62.4.4 bool prior_hessian::SymmetricBetaDist::check_params (double *beta*) [inline],[static]

Definition at line 101 of file SymmetricBetaDist.h.

Referenced by check_params_iter(), and param_ubound().

Here is the caller graph for this function:



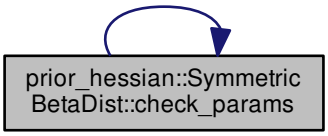
7.62.4.5 template<class Vec > static bool prior_hessian::SymmetricBetaDist::check_params (Vec & *p*) [inline], [static]

Definition at line 40 of file SymmetricBetaDist.h.

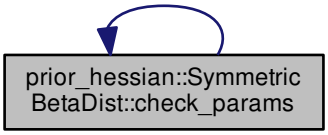
References check_params().

Referenced by check_params().

Here is the call graph for this function:



Here is the caller graph for this function:



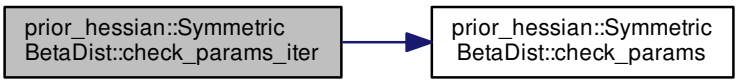
7.62.4.6 template<class IterT > bool prior_hessian::SymmetricBetaDist::check_params_iter (IterT & *params*) [static]

Definition at line 167 of file SymmetricBetaDist.h.

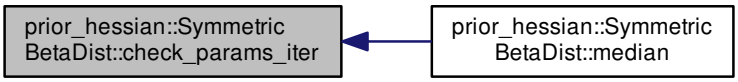
References check_params().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

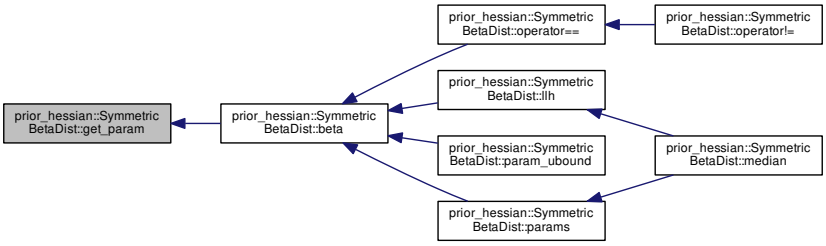


7.62.4.7 double prior_hessian::SymmetricBetaDist::get_param (int *idx*) const [inline]

Definition at line 107 of file SymmetricBetaDist.h.

Referenced by beta().

Here is the caller graph for this function:



7.62.4.8 double prior_hessian::SymmetricBetaDist::grad (double *x*) const [inline]

Definition at line 137 of file SymmetricBetaDist.h.

Referenced by median().

Here is the caller graph for this function:

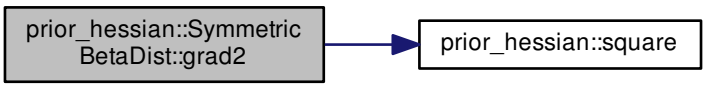
7.62.4.9 double prior_hessian::SymmetricBetaDist::grad2 (double *x*) const [inline]

Definition at line 143 of file SymmetricBetaDist.h.

References prior_hessian::square().

Referenced by median().

Here is the call graph for this function:



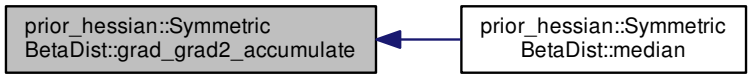
Here is the caller graph for this function:

7.62.4.10 void prior_hessian::SymmetricBetaDist::grad_grad2_accumulate (double *x*, double & *g*, double & *g2*) const [inline]

Definition at line 149 of file SymmetricBetaDist.h.

Referenced by median().

Here is the caller graph for this function:

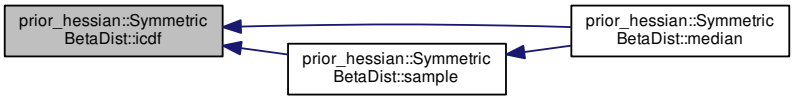


7.62.4.11 double prior_hessian::SymmetricBetaDist::icdf (double *u*) const

Definition at line 43 of file SymmetricBetaDist.cpp.

Referenced by median(), and sample().

Here is the caller graph for this function:

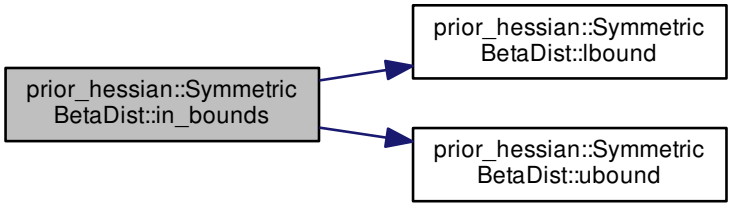


7.62.4.12 static bool prior_hessian::SymmetricBetaDist::in_bounds (double *u*) [inline], [static]

Definition at line 32 of file SymmetricBetaDist.h.

References lbound(), and ubound().

Here is the call graph for this function:



7.62.4.13 static constexpr double prior_hessian::SymmetricBetaDist::lbound ( ) [inline],[static]

Definition at line 30 of file SymmetricBetaDist.h.

Referenced by in_bounds().

Here is the caller graph for this function:

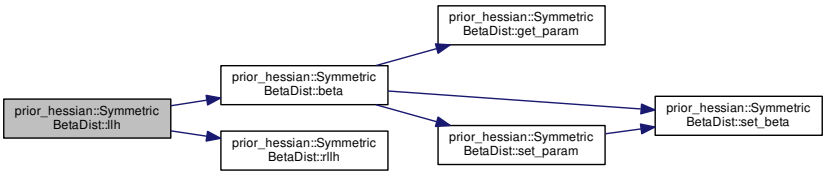
7.62.4.14 double prior_hessian::SymmetricBetaDist::llh (double *x*) const

Definition at line 55 of file SymmetricBetaDist.cpp.

References beta(), and rllh().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

7.62.4.15 double prior_hessian::SymmetricBetaDist::mean ( ) const [inline]

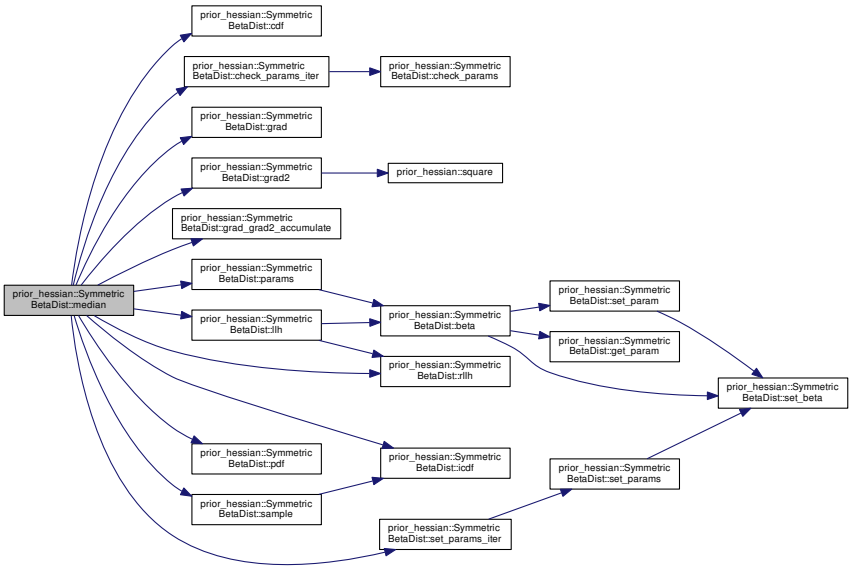
Definition at line 60 of file SymmetricBetaDist.h.

7.62.4.16 double prior_hessian::SymmetricBetaDist::median ( ) const [inline]

Definition at line 61 of file SymmetricBetaDist.h.

References cdf(), check_params_iter(), grad(), grad2(), grad_grad2_accumulate(), icdf(), llh(), params(), pdf(), rllh(), sample(), and set_params_iter().

Here is the call graph for this function:



7.62.4.17 static constexpr IdxT prior_hessian::UnivariateDist::num_dim ( ) [inline], [static], [inherited]

Definition at line 17 of file UnivariateDist.h.

7.62.4.18 static constexpr IdxT prior_hessian::SymmetricBetaDist::num_params ( ) [inline],[static]

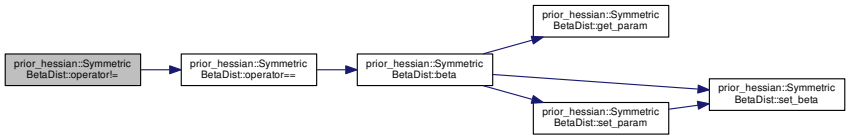
Definition at line 29 of file SymmetricBetaDist.h.

7.62.4.19 bool prior_hessian::SymmetricBetaDist::operator!= (const SymmetricBetaDist & *o*) const [inline]

Definition at line 58 of file SymmetricBetaDist.h.

References operator==().

Here is the call graph for this function:



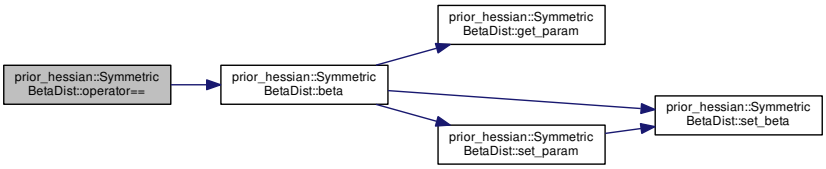
7.62.4.20 bool prior_hessian::SymmetricBetaDist::operator== (const SymmetricBetaDist & *o*) const [inline]

Definition at line 57 of file SymmetricBetaDist.h.

References beta().

Referenced by operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

7.62.4.21 static const VecT& prior_hessian::SymmetricBetaDist::param_lbound ( ) [inline], [static]

Definition at line 35 of file SymmetricBetaDist.h.

7.62.4.22 static const StringVecT& prior_hessian::SymmetricBetaDist::param_names ( ) [inline], [static]

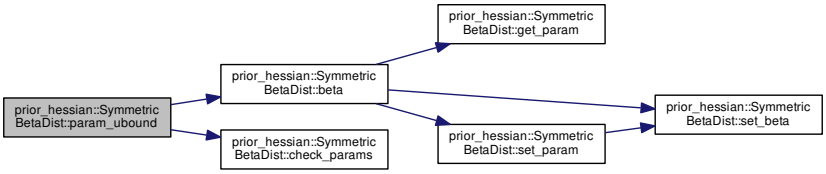
Definition at line 34 of file SymmetricBetaDist.h.

7.62.4.23 static const VecT& prior_hessian::SymmetricBetaDist::param_ubound ( ) [inline], [static]

Definition at line 36 of file SymmetricBetaDist.h.

References beta(), and check_params().

Here is the call graph for this function:



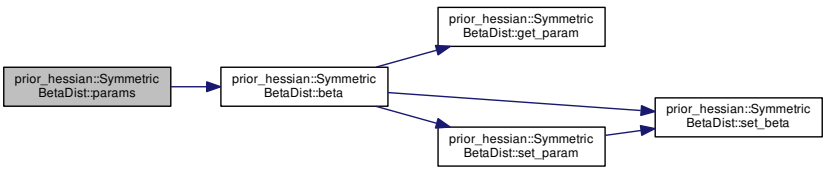
7.62.4.24 NparamsVecT prior_hessian::SymmetricBetaDist::params ( ) const [inline]

Definition at line 52 of file SymmetricBetaDist.h.

References beta().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:

7.62.4.25 double prior_hessian::SymmetricBetaDist::pdf (double *x*) const

Definition at line 50 of file SymmetricBetaDist.cpp.

Referenced by median().

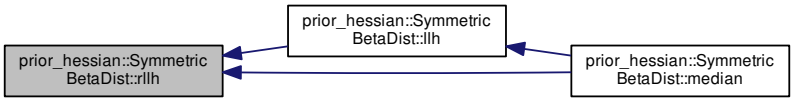
Here is the caller graph for this function:

7.62.4.26 double prior_hessian::SymmetricBetaDist::rllh (double *x*) const [inline]

Definition at line 131 of file SymmetricBetaDist.h.

Referenced by llh(), and median().

Here is the caller graph for this function:



7.62.4.27 template<class RngT > double prior_hessian::SymmetricBetaDist::sample (RngT & *rng*) const

Definition at line 159 of file SymmetricBetaDist.h.

References icdf().

Referenced by median().

Here is the call graph for this function:

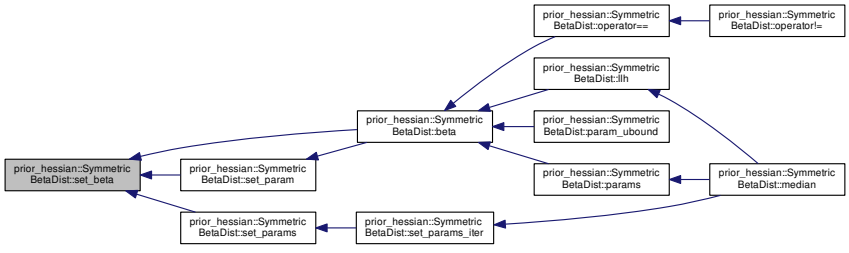
Here is the caller graph for this function:

7.62.4.28 void prior_hessian::SymmetricBetaDist::set_beta (double *val*)

Definition at line 30 of file SymmetricBetaDist.cpp.

Referenced by beta(), set_param(), and set_params().

Here is the caller graph for this function:



7.62.4.29 void prior_hessian::SymmetricBetaDist::set_param (int *idx*, double *val*) [inline]

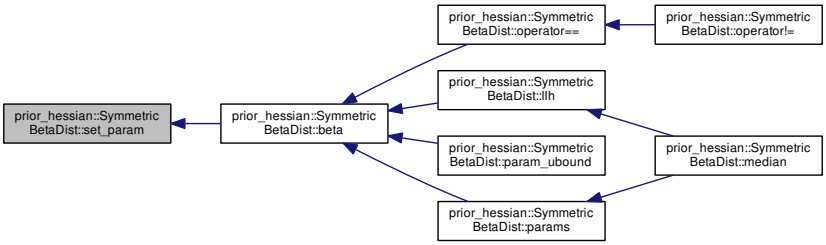
Definition at line 119 of file SymmetricBetaDist.h.

References set_beta().

Referenced by beta().

Here is the call graph for this function:

Here is the caller graph for this function:



7.62.4.30 void prior_hessian::SymmetricBetaDist::set_params (double *beta*) [inline]

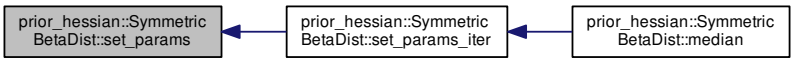
Definition at line 53 of file SymmetricBetaDist.h.

References set_beta().

Referenced by set_params_iter().

Here is the call graph for this function:

Here is the caller graph for this function:

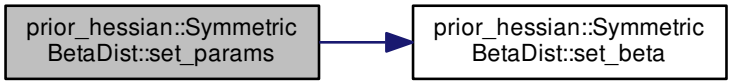


7.62.4.31 template<class Vec > void prior_hessian::SymmetricBetaDist::set_params (const Vec & *p*) [inline]

Definition at line 56 of file SymmetricBetaDist.h.

References set_beta().

Here is the call graph for this function:



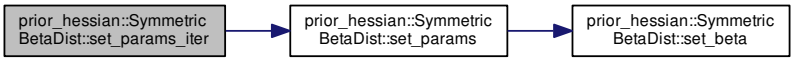
7.62.4.32 template<class IterT > void prior_hessian::SymmetricBetaDist::set_params_iter (IterT & *params*)

Definition at line 173 of file SymmetricBetaDist.h.

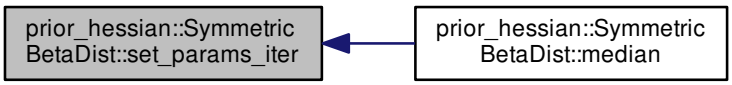
References set_params().

Referenced by median().

Here is the call graph for this function:



Here is the caller graph for this function:



7.62.4.33 static constexpr double prior_hessian::SymmetricBetaDist::ubound ( ) [inline], [static]

Definition at line 31 of file SymmetricBetaDist.h.

Referenced by in_bounds().

Here is the caller graph for this function:

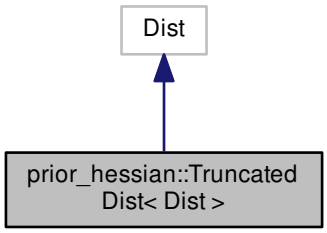
The documentation for this class was generated from the following files:

- SymmetricBetaDist.h
- SymmetricBetaDist.cpp

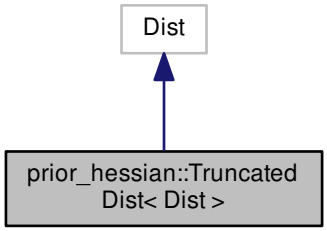
7.63 prior_hessian::TruncatedDist< Dist > Class Template Reference

```
#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩
TruncatedDist.h>
```

Inheritance diagram for prior_hessian::TruncatedDist< Dist >:



Collaboration diagram for prior_hessian::TruncatedDist< Dist >:



Public Member Functions

- TruncatedDist ()
- TruncatedDist (double lbound, double ubound)
- template<typename = meta::EnableIfNotIsSelfT<Dist,TruncatedDist>>
  TruncatedDist (const Dist &dist)
- template<typename = meta::EnableIfNotIsSelfT<Dist,TruncatedDist>>
  TruncatedDist (Dist &&dist)
- TruncatedDist (const Dist &dist, double lbound, double ubound)
- TruncatedDist (Dist &&dist, double lbound, double ubound)
- double lbound () const
- double ubound () const
- bool truncated () const
- bool operator== (const TruncatedDist< Dist > &o) const
- bool operator!= (const TruncatedDist< Dist > &o) const
- void set_bounds (double lbound, double ubound)

- void set_lbound (double lbound)
- void set_ubound (double ubound)
- double mean () const
- double median () const
- double cdf (double x) const
- double pdf (double x) const
- double icdf (double u) const
- double llh (double x) const
- template<class RngT >
  double sample (RngT &rng) const

Static Public Member Functions

- static constexpr IdxT num_params ()
- static double global_lbound ()
- static double global_ubound ()

Static Public Attributes

- static constexpr const double min_bounds_pdf_integral = 1.0e-8

Protected Attributes

- double _truncated_lbound
- double _truncated_ubound
- bool _truncated = false
- double lbound_cdf
- double bounds_pdf_integral
- double llh_truncation_const

7.63.1 Detailed Description

template<class Dist>
class prior_hessian::TruncatedDist< Dist >

Definition at line 22 of file TruncatedDist.h.

7.63.2 Constructor & Destructor Documentation

7.63.2.1 template<class Dist> prior_hessian::TruncatedDist< Dist >::TruncatedDist ( ) [inline]

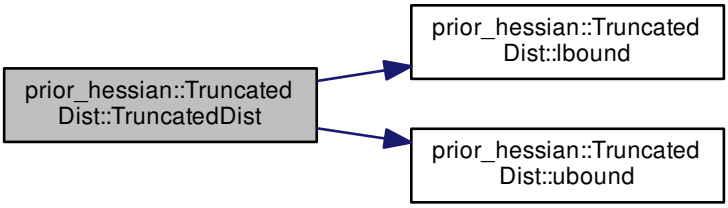
Definition at line 30 of file TruncatedDist.h.

7.63.2.2 template<class Dist> prior_hessian::TruncatedDist< Dist >::TruncatedDist (double *lbound,* double *ubound*) [inline]

Definition at line 31 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::lbound(), and prior_hessian::TruncatedDist< Dist >::ubound().

Here is the call graph for this function:



7.63.2.3 template<class Dist> template<typename = meta::EnableIfNotIsSelfT<Dist,TruncatedDist>> prior_hessian::TruncatedDist< Dist >::TruncatedDist (const Dist & *dist*) [inline]

Definition at line 34 of file TruncatedDist.h.

7.63.2.4 template<class Dist> template<typename = meta::EnableIfNotIsSelfT<Dist,TruncatedDist>> prior_hessian::TruncatedDist< Dist >::TruncatedDist (Dist && *dist*) [inline]

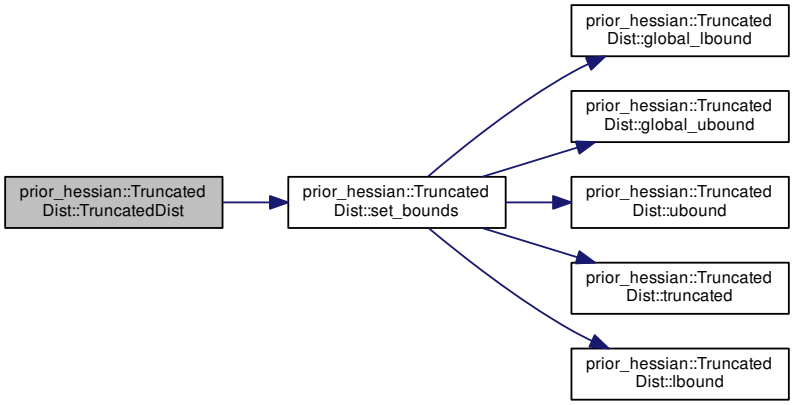
Definition at line 37 of file TruncatedDist.h.

7.63.2.5 template<class Dist> prior_hessian::TruncatedDist< Dist >::TruncatedDist (const Dist & *dist,* double *lbound,* double *ubound*) [inline]

Definition at line 39 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::set_bounds().

Here is the call graph for this function:

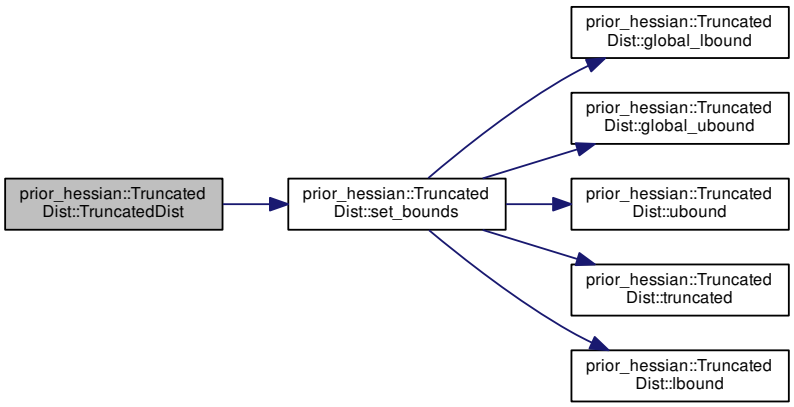


7.63.2.6 template<class Dist> prior_hessian::TruncatedDist< Dist >::TruncatedDist (Dist && *dist*, double *lbound*, double *ubound*) [inline]

Definition at line 43 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::set_bounds().

Here is the call graph for this function:



7.63.3 Member Function Documentation

7.63.3.1 template<class Dist > double prior_hessian::TruncatedDist< Dist >::cdf (double *x*) const

Definition at line 135 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::bounds_pdf_integral, and prior_hessian::TruncatedDist< Dist >::lbound_cdf.

Referenced by prior_hessian::TruncatedDist< Dist >::median().

Here is the caller graph for this function:

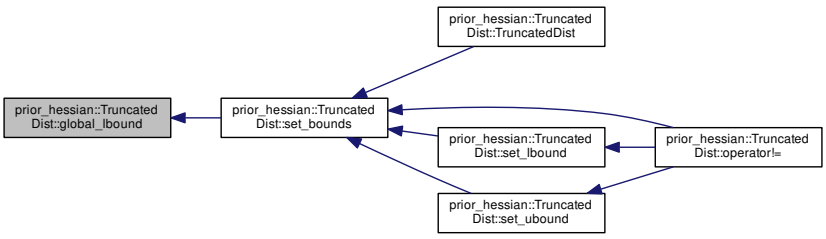
7.63.3.2 template<class Dist> static double prior_hessian::TruncatedDist< Dist >::global_lbound ( ) [inline], [static]

minimum allowable delta in cdf for a valid truncation

Definition at line 27 of file TruncatedDist.h.

Referenced by prior_hessian::TruncatedDist< Dist >::set_bounds().

Here is the caller graph for this function:

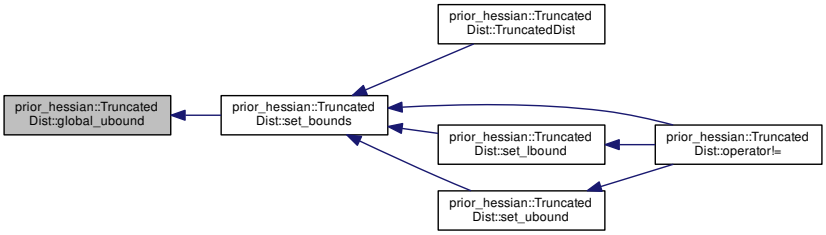


7.63.3.3 template<class Dist> static double prior_hessian::TruncatedDist< Dist >::global_ubound ( ) [inline], [static]

Definition at line 28 of file TruncatedDist.h.

Referenced by prior_hessian::TruncatedDist< Dist >::set_bounds().

Here is the caller graph for this function:



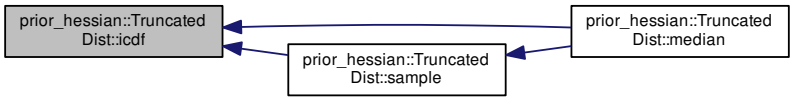
7.63.3.4 template<class Dist > double prior_hessian::TruncatedDist< Dist >::icdf (double *u*) const

Definition at line 141 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::bounds_pdf_integral, and prior_hessian::TruncatedDist< Dist >::lbound_cdf.

Referenced by prior_hessian::TruncatedDist< Dist >::median(), and prior_hessian::TruncatedDist< Dist >::sample().

Here is the caller graph for this function:



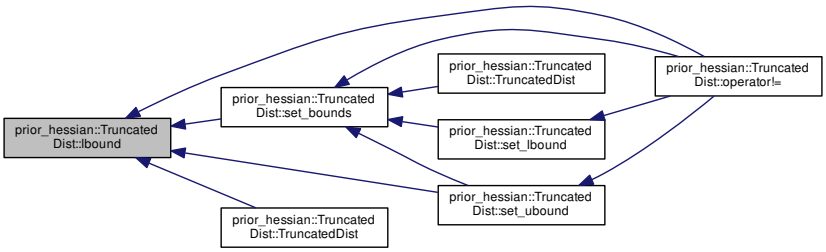
7.63.3.5 template<class Dist> double prior_hessian::TruncatedDist< Dist >::lbound ( ) const [inline]

Definition at line 47 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::_truncated_lbound.

Referenced by prior_hessian::TruncatedDist< Dist >::operator!=(), prior_hessian::TruncatedDist< Dist >::set_bounds(), prior_hessian::TruncatedDist< Dist >::set_ubound(), and prior_hessian::TruncatedDist< Dist >::TruncatedDist().

Here is the caller graph for this function:



7.63.3.6 template<class Dist > double prior_hessian::TruncatedDist< Dist >::llh (double *x*) const

Definition at line 153 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::llh_truncation_const.

Referenced by prior_hessian::TruncatedDist< Dist >::median().

Here is the caller graph for this function:

7.63.3.7 template<class Dist> double prior_hessian::TruncatedDist< Dist >::mean ( ) const [inline]

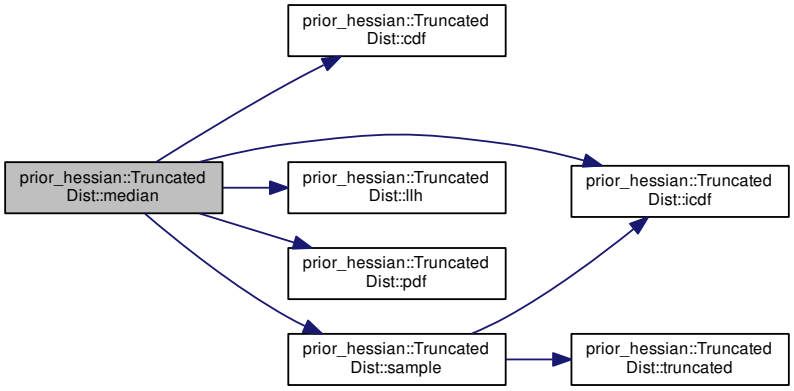
Definition at line 62 of file TruncatedDist.h.

7.63.3.8 template<class Dist> double prior_hessian::TruncatedDist< Dist >::median ( ) const [inline]

Definition at line 63 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::bounds_pdf_integral, prior_hessian::TruncatedDist< Dist >::cdf(), prior_hessian::TruncatedDist< Dist >::icdf(), prior_hessian::TruncatedDist< Dist >::lbound_cdf, prior_hessian::TruncatedDist< Dist >::llh(), prior_hessian::TruncatedDist< Dist >::pdf(), and prior_hessian::TruncatedDist< Dist >::sample().

Here is the call graph for this function:



7.63.3.9 template<class Dist> static constexpr IdxT prior_hessian::TruncatedDist< Dist >::num_params ( ) [inline], [static]

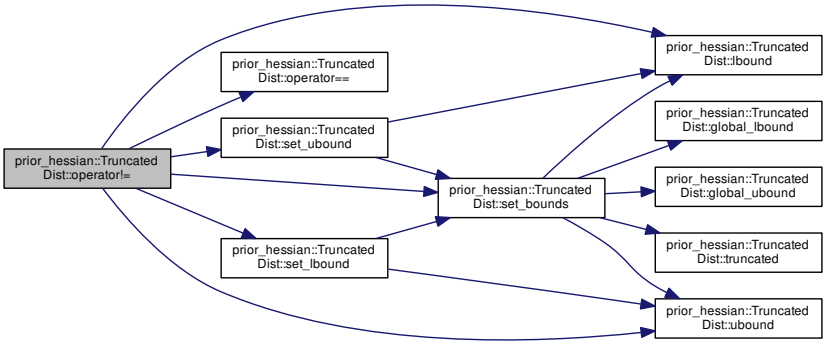
Definition at line 25 of file TruncatedDist.h.

7.63.3.10 template<class Dist> bool prior_hessian::TruncatedDist< Dist >::operator!= (const TruncatedDist< Dist > & *o*) const [inline]

Definition at line 56 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::lbound(), prior_hessian::TruncatedDist< Dist >::operator==(), prior_hessian::TruncatedDist< Dist >::set_bounds(), prior_hessian::TruncatedDist< Dist >::set_lbound(), prior_↩ hessian::TruncatedDist< Dist >::set_ubound(), and prior_hessian::TruncatedDist< Dist >::ubound().

Here is the call graph for this function:



7.63.3.11 template<class Dist> bool prior_hessian::TruncatedDist< Dist >::operator== (const TruncatedDist< Dist > & *o*) const [inline]

Definition at line 50 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::_truncated_lbound, and prior_hessian::TruncatedDist< Dist >::_↩ truncated_ubound.

Referenced by prior_hessian::TruncatedDist< Dist >::operator!=().

Here is the caller graph for this function:

7.63.3.12 template<class Dist > double prior_hessian::TruncatedDist< Dist >::pdf (double *x*) const

Definition at line 147 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::bounds_pdf_integral.

Referenced by prior_hessian::TruncatedDist< Dist >::median().

Here is the caller graph for this function:

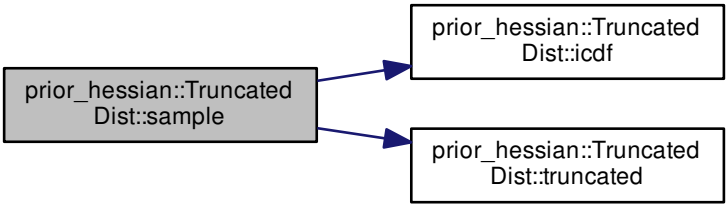
7.63.3.13 template<class Dist > template<class RngT > double prior_hessian::TruncatedDist< Dist >::sample (RngT & *rng*) const

Definition at line 166 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::icdf(), and prior_hessian::TruncatedDist< Dist >::truncated().

Referenced by prior_hessian::TruncatedDist< Dist >::median().

Here is the call graph for this function:



Here is the caller graph for this function:

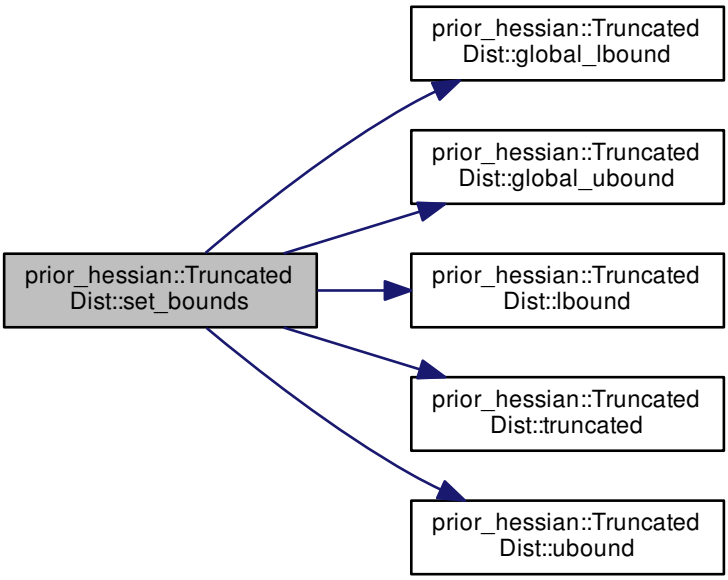
7.63.3.14 template<class Dist > void prior_hessian::TruncatedDist< Dist >::set_bounds (double *lbound,* double *ubound*)

Definition at line 82 of file TruncatedDist.h.

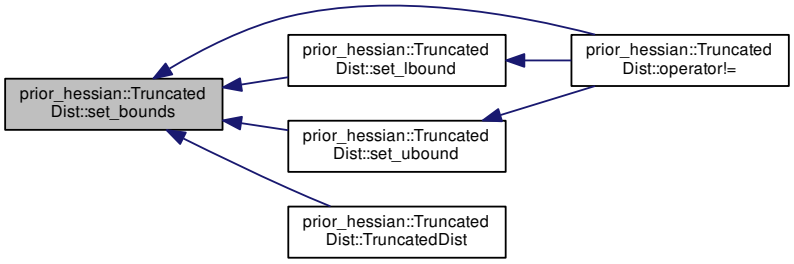
References prior_hessian::TruncatedDist< Dist >::_truncated, prior_hessian::TruncatedDist< Dist >::_truncated_lbound, prior_hessian::TruncatedDist< Dist >::_truncated_ubound, prior_hessian::TruncatedDist< Dist >::bounds_pdf_integral, prior_hessian::TruncatedDist< Dist >::global_lbound(), prior_hessian::TruncatedDist< Dist >::global_ubound(), prior_hessian::TruncatedDist< Dist >::lbound(), prior_hessian::TruncatedDist< Dist >::lbound_cdf, prior_hessian::TruncatedDist< Dist >::llh_truncation_const, prior_hessian::TruncatedDist< Dist >::min_bounds_pdf_integral, prior_hessian::TruncatedDist< Dist >::truncated(), and prior_hessian::TruncatedDist< Dist >::ubound().

Referenced by prior_hessian::TruncatedDist< Dist >::operator!=(), prior_hessian::TruncatedDist< Dist >::set_lbound(), prior_hessian::TruncatedDist< Dist >::set_ubound(), and prior_hessian::TruncatedDist< Dist >::TruncatedDist().

Here is the call graph for this function:



Here is the caller graph for this function:



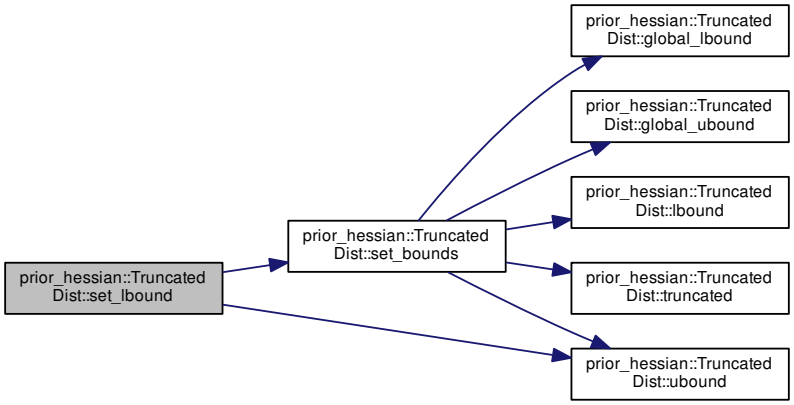
7.63.3.15 template<class Dist > void prior_hessian::TruncatedDist< Dist >::set_lbound (double *lbound*)

Definition at line 123 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::set_bounds(), and prior_hessian::TruncatedDist< Dist >::ubound().

Referenced by prior_hessian::TruncatedDist< Dist >::operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

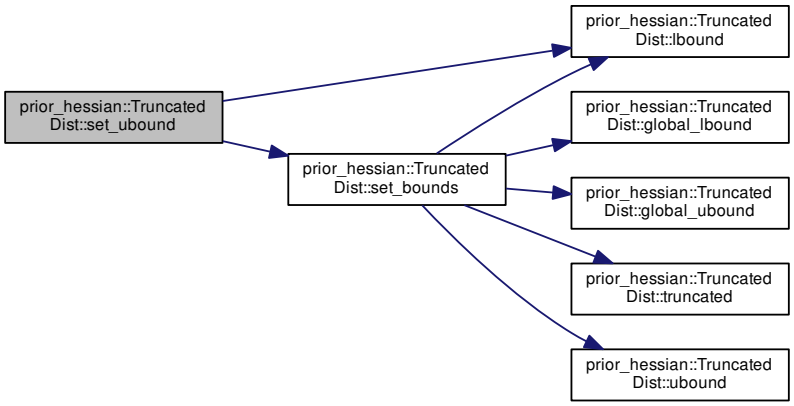
7.63.3.16 template<class Dist > void prior_hessian::TruncatedDist< Dist >::set_ubound (double *ubound*)

Definition at line 129 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::lbound(), and prior_hessian::TruncatedDist< Dist >::set_bounds().

Referenced by prior_hessian::TruncatedDist< Dist >::operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

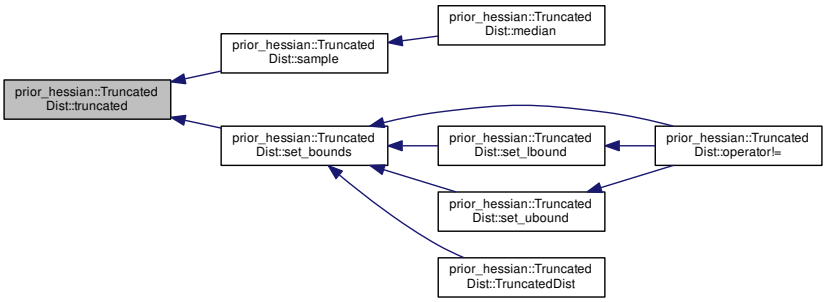
7.63.3.17 template<class Dist> bool prior_hessian::TruncatedDist< Dist >::truncated ( ) const [inline]

Definition at line 49 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::_truncated.

Referenced by prior_hessian::TruncatedDist< Dist >::sample(), and prior_hessian::TruncatedDist< Dist >::set_↩ bounds().

Here is the caller graph for this function:



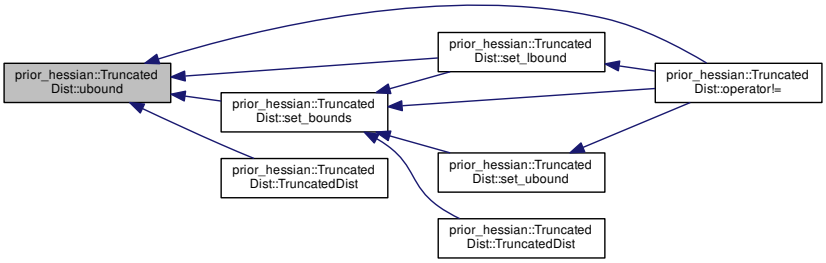
7.63.3.18 template<class Dist> double prior_hessian::TruncatedDist< Dist >::ubound ( ) const [inline]

Definition at line 48 of file TruncatedDist.h.

References prior_hessian::TruncatedDist< Dist >::_truncated_ubound.

Referenced by prior_hessian::TruncatedDist< Dist >::operator!=(), prior_hessian::TruncatedDist< Dist >::set_↩ bounds(), prior_hessian::TruncatedDist< Dist >::set_lbound(), and prior_hessian::TruncatedDist< Dist >::Truncated↩ Dist().

Here is the caller graph for this function:



7.63.4 Member Data Documentation

7.63.4.1 template<class Dist> bool prior_hessian::TruncatedDist< Dist >::_truncated = false [protected]

Definition at line 74 of file TruncatedDist.h.

Referenced by prior_hessian::TruncatedDist< Dist >::set_bounds(), and prior_hessian::TruncatedDist< Dist >↩ ::truncated().

7.63.4.2 template<class Dist> double prior_hessian::TruncatedDist< Dist >::_truncated_lbound [protected]

Definition at line 72 of file TruncatedDist.h.

Referenced by prior_hessian::TruncatedDist< Dist >::lbound(), prior_hessian::TruncatedDist< Dist >::operator==(), and prior_hessian::TruncatedDist< Dist >::set_bounds().

7.63.4.3 template<class Dist> double prior_hessian::TruncatedDist< Dist >::_truncated_ubound [protected]

Definition at line 73 of file TruncatedDist.h.

Referenced by prior_hessian::TruncatedDist< Dist >::operator==(), prior_hessian::TruncatedDist< Dist >::set_↩bounds(), and prior_hessian::TruncatedDist< Dist >::ubound().

7.63.4.4 template<class Dist> double prior_hessian::TruncatedDist< Dist >::bounds_pdf_integral [protected]

Definition at line 77 of file TruncatedDist.h.

Referenced by prior_hessian::TruncatedDist< Dist >::cdf(), prior_hessian::TruncatedDist< Dist >::icdf(), prior↩_hessian::TruncatedDist< Dist >::median(), prior_hessian::TruncatedDist< Dist >::pdf(), and prior_hessian::↩TruncatedDist< Dist >::set_bounds().

7.63.4.5 template<class Dist> double prior_hessian::TruncatedDist< Dist >::lbound_cdf [protected]

Definition at line 76 of file TruncatedDist.h.

Referenced by prior_hessian::TruncatedDist< Dist >::cdf(), prior_hessian::TruncatedDist< Dist >::icdf(), prior_↩hessian::TruncatedDist< Dist >::median(), and prior_hessian::TruncatedDist< Dist >::set_bounds().

7.63.4.6 template<class Dist> double prior_hessian::TruncatedDist< Dist >::llh_truncation_const [protected]

Definition at line 78 of file TruncatedDist.h.

Referenced by prior_hessian::TruncatedDist< Dist >::llh(), and prior_hessian::TruncatedDist< Dist >::set_bounds().

7.63.4.7 template<class Dist> constexpr const double prior_hessian::TruncatedDist< Dist >::min_bounds_pdf_integral = 1.0e-8 [static]

Definition at line 26 of file TruncatedDist.h.

Referenced by prior_hessian::TruncatedDist< Dist >::set_bounds().

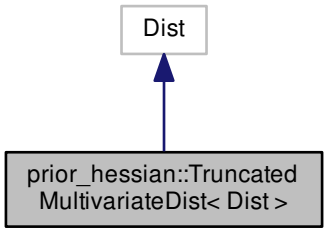
The documentation for this class was generated from the following file:

- TruncatedDist.h

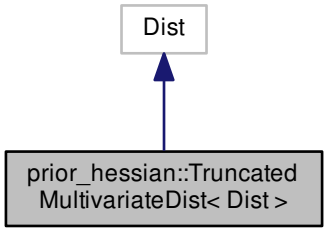
7.64 prior_hessian::TruncatedMultivariateDist< Dist > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/TruncatedMultivariateDist.h>`

Inheritance diagram for prior_hessian::TruncatedMultivariateDist< Dist >:



Collaboration diagram for prior_hessian::TruncatedMultivariateDist< Dist >:



Public Member Functions

- TruncatedMultivariateDist ()
- template<class Vec >
  TruncatedMultivariateDist (Vec &&lbound, Vec &&ubound)
- template<typename = meta::EnableIfNotIsSelfT<Dist,TruncatedMultivariateDist>>
  TruncatedMultivariateDist (const Dist &dist)
- template<typename = meta::EnableIfNotIsSelfT<Dist,TruncatedMultivariateDist>>
  TruncatedMultivariateDist (Dist &&dist)
- template<class Vec >
  TruncatedMultivariateDist (const Dist &dist, Vec &&lbound, Vec &&ubound)

- template<class Vec >
  TruncatedMultivariateDist (Dist &&dist, Vec &&lbound, Vec &&ubound)
- const NdimVecT & lbound () const
- const NdimVecT & ubound () const
- template<class Vec >
  bool in_bounds (const Vec &u) const
- bool truncated () const
- bool operator== (const TruncatedMultivariateDist< Dist > &o) const
- bool operator!= (const TruncatedMultivariateDist< Dist > &o) const
- template<class Vec , class Vec2 >
  void set_bounds (const Vec &lbound, const Vec2 &ubound)
- template<class Vec >
  void set_lbound (const Vec &lbound)
- template<class Vec >
  void set_ubound (const Vec &ubound)
- double mean () const
- template<class Vec >
  double cdf (const Vec &x) const
- template<class Vec >
  double pdf (const Vec &x) const
- template<class Vec >
  double llh (const Vec &x) const
- template<class RngT >
  NdimVecT sample (RngT &rng) const

Static Public Member Functions

- static const NdimVecT & global_lbound ()
- static const NdimVecT & global_ubound ()

Static Public Attributes

- static constexpr const double min_bounds_pdf_integral = 1.0e-8

Protected Member Functions

- double compute_truncated_pdf_integral (const NdimVecT &lbound, const NdimVecT &ubound, double lbound↩
  _cdf) const

Protected Attributes

- NdimVecT _truncated_lbound
- NdimVecT _truncated_ubound
- bool _truncated = false
- double lbound_cdf
- double bounds_pdf_integral
- double llh_truncation_const

7.64.1 Detailed Description

template<class Dist>
class prior_hessian::TruncatedMultivariateDist< Dist >

Definition at line 60 of file TruncatedMultivariateDist.h.

7.64.2 Constructor & Destructor Documentation

7.64.2.1 template<class Dist> prior_hessian::TruncatedMultivariateDist< Dist >::TruncatedMultivariateDist ( ) [inline]

minimum allowabale integral of pdf for a valid truncation

Definition at line 66 of file TruncatedMultivariateDist.h.

7.64.2.2 template<class Dist> template<class Vec > prior_hessian::TruncatedMultivariateDist< Dist >::TruncatedMultivariateDist (Vec && *lbound,* Vec && *ubound*) [inline]

Definition at line 69 of file TruncatedMultivariateDist.h.

7.64.2.3 template<class Dist> template<typename = meta::EnableIfNotIsSelfT<Dist,TruncatedMultivariateDist>> prior_hessian::TruncatedMultivariateDist< Dist >::TruncatedMultivariateDist (const Dist & *dist*) [inline]

Definition at line 73 of file TruncatedMultivariateDist.h.

7.64.2.4 template<class Dist> template<typename = meta::EnableIfNotIsSelfT<Dist,TruncatedMultivariateDist>> prior_hessian::TruncatedMultivariateDist< Dist >::TruncatedMultivariateDist (Dist && *dist*) [inline]

Definition at line 76 of file TruncatedMultivariateDist.h.

7.64.2.5 template<class Dist> template<class Vec > prior_hessian::TruncatedMultivariateDist< Dist >::TruncatedMultivariateDist (const Dist & *dist,* Vec && *lbound,* Vec && *ubound*) [inline]

Definition at line 79 of file TruncatedMultivariateDist.h.

7.64.2.6 template<class Dist> template<class Vec > prior_hessian::TruncatedMultivariateDist< Dist >::TruncatedMultivariateDist (Dist && *dist,* Vec && *lbound,* Vec && *ubound*) [inline]

Definition at line 86 of file TruncatedMultivariateDist.h.

7.64.3 Member Function Documentation

7.64.3.1 template<class Dist > template<class Vec > double prior_hessian::TruncatedMultivariateDist< Dist >::cdf (const Vec & *x*) const

Definition at line 233 of file TruncatedMultivariateDist.h.

7.64.3.2 template<class Dist > double prior_hessian::TruncatedMultivariateDist< Dist >::compute_truncated_pdf_integral (const NdimVecT & *lbound,* const NdimVecT & *ubound,* double *lbound_cdf*) const [protected]

Definition at line 147 of file TruncatedMultivariateDist.h.

7.64.3.3 template<class Dist> static const NdimVecT& prior_hessian::TruncatedMultivariateDist< Dist >::global_lbound ( ) [inline], [static]

Definition at line 96 of file TruncatedMultivariateDist.h.

7.64.3.4 template<class Dist> static const NdimVecT& prior_hessian::TruncatedMultivariateDist< Dist >::global_ubound ( ) [inline], [static]

Definition at line 97 of file TruncatedMultivariateDist.h.

7.64.3.5 template<class Dist> template<class Vec > bool prior_hessian::TruncatedMultivariateDist< Dist >::in_bounds (const Vec & *u*) const [inline]

Definition at line 95 of file TruncatedMultivariateDist.h.

7.64.3.6 template<class Dist> const NdimVecT& prior_hessian::TruncatedMultivariateDist< Dist >::lbound ( ) const [inline]

Definition at line 92 of file TruncatedMultivariateDist.h.

7.64.3.7 template<class Dist > template<class Vec > double prior_hessian::TruncatedMultivariateDist< Dist >::llh (const Vec & *x*) const

Definition at line 248 of file TruncatedMultivariateDist.h.

7.64.3.8 template<class Dist> double prior_hessian::TruncatedMultivariateDist< Dist >::mean ( ) const [inline]

Definition at line 114 of file TruncatedMultivariateDist.h.

References prior_hessian::mcmc::MCMCData< Ndim >::sample.

7.64.3.9 template<class Dist> bool prior_hessian::TruncatedMultivariateDist< Dist >::operator!= (const TruncatedMultivariateDist< Dist > & *o*) const [inline]

Definition at line 105 of file TruncatedMultivariateDist.h.

7.64.3.10 template<class Dist> bool prior_hessian::TruncatedMultivariateDist< Dist >::operator== (const TruncatedMultivariateDist< Dist > & *o*) const [inline]

Definition at line 99 of file TruncatedMultivariateDist.h.

References prior_hessian::TruncatedMultivariateDist< Dist >::_truncated_lbound, and prior_hessian::Truncated↩ MultivariateDist< Dist >::_truncated_ubound.

7.64.3.11 template<class Dist > template<class Vec > double prior_hessian::TruncatedMultivariateDist< Dist >::pdf (const Vec & *x*) const

Definition at line 241 of file TruncatedMultivariateDist.h.

7.64.3.12 template<class Dist > template<class RngT > TruncatedMultivariateDist< Dist >::NdimVecT prior_hessian::TruncatedMultivariateDist< Dist >::sample (RngT & *rng*) const

Definition at line 256 of file TruncatedMultivariateDist.h.

References prior_hessian::mcmc::MCMCData< Ndim >::rllh, and prior_hessian::mcmc::MCMCData< Ndim >::sample.

7.64.3.13 template<class Dist > template<class Vec , class Vec2 > void prior_hessian::TruncatedMultivariateDist< Dist >::set_bounds (const Vec & *lbound,* const Vec2 & *ubound*)

Definition at line 179 of file TruncatedMultivariateDist.h.

7.64.3.14 template<class Dist > template<class Vec > void prior_hessian::TruncatedMultivariateDist< Dist >::set_lbound (const Vec & *lbound*)

Definition at line 219 of file TruncatedMultivariateDist.h.

7.64.3.15 template<class Dist > template<class Vec > void prior_hessian::TruncatedMultivariateDist< Dist >::set_ubound (const Vec & *ubound*)

Definition at line 226 of file TruncatedMultivariateDist.h.

7.64.3.16 template<class Dist> bool prior_hessian::TruncatedMultivariateDist< Dist >::truncated ( ) const [inline]

Definition at line 98 of file TruncatedMultivariateDist.h.

7.64.3.17 template<class Dist> const NdimVecT& prior_hessian::TruncatedMultivariateDist< Dist >::ubound ( ) const [inline]

Definition at line 93 of file TruncatedMultivariateDist.h.

7.64.4 Member Data Documentation

7.64.4.1 template<class Dist> bool prior_hessian::TruncatedMultivariateDist< Dist >::_truncated = false [protected]

Definition at line 128 of file TruncatedMultivariateDist.h.

7.64.4.2 template<class Dist> NdimVecT prior_hessian::TruncatedMultivariateDist< Dist >::_truncated_lbound [protected]

Definition at line 126 of file TruncatedMultivariateDist.h.

Referenced by prior_hessian::TruncatedMultivariateDist< Dist >::operator==().

7.64.4.3 template<class Dist> NdimVecT prior_hessian::TruncatedMultivariateDist< Dist >::_truncated_ubound [protected]

Definition at line 127 of file TruncatedMultivariateDist.h.

Referenced by prior_hessian::TruncatedMultivariateDist< Dist >::operator==().

7.64.4.4 template<class Dist> double prior_hessian::TruncatedMultivariateDist< Dist >::bounds_pdf_integral [protected]

Definition at line 132 of file TruncatedMultivariateDist.h.

7.64.4.5 template<class Dist> double prior_hessian::TruncatedMultivariateDist< Dist >::lbound_cdf [protected]

Definition at line 131 of file TruncatedMultivariateDist.h.

7.64.4.6 template<class Dist> double prior_hessian::TruncatedMultivariateDist< Dist >::llh_truncation_const [protected]

Definition at line 133 of file TruncatedMultivariateDist.h.

7.64.4.7 template<class Dist> constexpr const double prior_hessian::TruncatedMultivariateDist< Dist >::min_bounds_pdf_integral = 1.0e-8 [static]

Definition at line 64 of file TruncatedMultivariateDist.h.

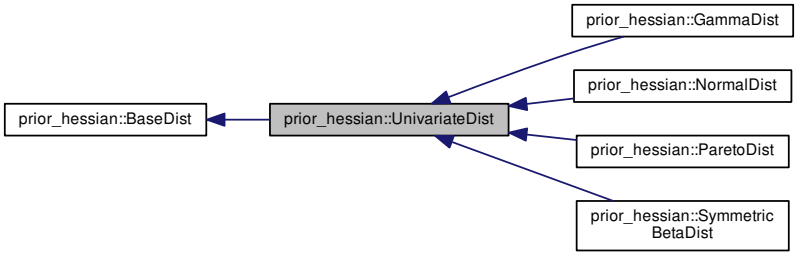
The documentation for this class was generated from the following file:

- TruncatedMultivariateDist.h

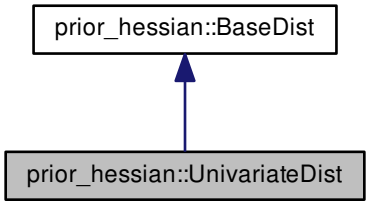
7.65 prior_hessian::UnivariateDist Class Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ UnivariateDist.h>`

Inheritance diagram for prior_hessian::UnivariateDist:



Collaboration diagram for prior_hessian::UnivariateDist:



Public Member Functions

- UnivariateDist ()

Static Public Member Functions

- static constexpr IdxT num_dim ()

Static Protected Member Functions

- static void check_bounds (double lbound, double ubound)

7.65.1 Detailed Description

Definition at line 15 of file UnivariateDist.h.

7.65.2 Constructor & Destructor Documentation

7.65.2.1 prior_hessian::UnivariateDist::UnivariateDist ( ) [inline]

Definition at line 19 of file UnivariateDist.h.

References check_bounds().

Here is the call graph for this function:

7.65.3 Member Function Documentation

7.65.3.1 void prior_hessian::UnivariateDist::check_bounds (double *lbound,* double *ubound*) [static],[protected]

Definition at line 17 of file UnivariateDist.cpp.

Referenced by UnivariateDist().

Here is the caller graph for this function:

7.65.3.2 static constexpr IdxT prior_hessian::UnivariateDist::num_dim ( ) [inline], [static]

Definition at line 17 of file UnivariateDist.h.

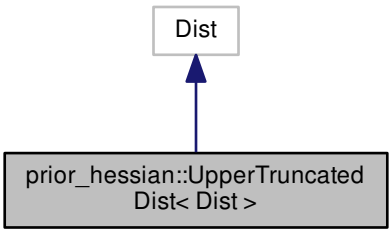
The documentation for this class was generated from the following files:

- UnivariateDist.h
- UnivariateDist.cpp

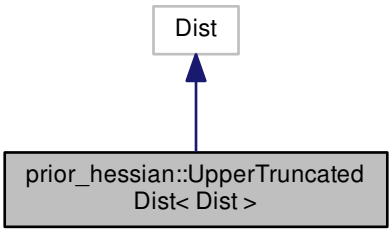
7.66 prior_hessian::UpperTruncatedDist< Dist > Class Template Reference

`#include </home/travis/build/markjolah/PriorHessian/include/PriorHessian/↩ UpperTruncatedDist.h>`

Inheritance diagram for prior_hessian::UpperTruncatedDist< Dist >:



Collaboration diagram for prior_hessian::UpperTruncatedDist< Dist >:



Public Member Functions

- UpperTruncatedDist ()
- UpperTruncatedDist (double ubound)
- template<typename = meta::EnableIfNotIsSelfT<Dist,UpperTruncatedDist>>
  UpperTruncatedDist (const Dist &dist)
- template<typename = meta::EnableIfNotIsSelfT<Dist,UpperTruncatedDist>>
  UpperTruncatedDist (Dist &&dist)
- UpperTruncatedDist (const Dist &dist, double ubound)
- UpperTruncatedDist (Dist &&dist, double ubound)
- double ubound () const
- bool truncated () const
- bool operator== (const UpperTruncatedDist< Dist > &o) const
- bool operator!= (const UpperTruncatedDist< Dist > &o) const
- void set_bounds (double lbound, double ubound)
- void set_lbound (double ubound)
- void set_ubound (double ubound)
- double mean () const
- double median () const
- double cdf (double x) const
- double pdf (double x) const
- double icdf (double u) const
- double llh (double x) const
- template<class RngT >
  double sample (RngT &rng) const

Static Public Member Functions

- static double global_ubound ()

7.66.1 Detailed Description

template<class Dist>
class prior_hessian::UpperTruncatedDist< Dist >

Definition at line 22 of file UpperTruncatedDist.h.

7.66.2 Constructor & Destructor Documentation

7.66.2.1 template<class Dist> prior_hessian::UpperTruncatedDist< Dist >::UpperTruncatedDist ( ) [inline]

Definition at line 25 of file UpperTruncatedDist.h.

7.66.2.2 template<class Dist> prior_hessian::UpperTruncatedDist< Dist >::UpperTruncatedDist (double *ubound*) [inline],[explicit]

Definition at line 26 of file UpperTruncatedDist.h.

References prior_hessian::UpperTruncatedDist< Dist >::ubound().

Here is the call graph for this function:

7.66.2.3 template<class Dist> template<typename = meta::EnableIfNotIsSelfT<Dist,UpperTruncatedDist>> prior_hessian::UpperTruncatedDist< Dist >::UpperTruncatedDist (const Dist & *dist*) [inline]

Definition at line 29 of file UpperTruncatedDist.h.

7.66.2.4 template<class Dist> template<typename = meta::EnableIfNotIsSelfT<Dist,UpperTruncatedDist>> prior_hessian::UpperTruncatedDist< Dist >::UpperTruncatedDist (Dist && *dist*) [inline]

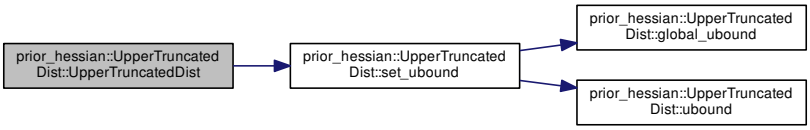
Definition at line 32 of file UpperTruncatedDist.h.

7.66.2.5 template<class Dist> prior_hessian::UpperTruncatedDist< Dist >::UpperTruncatedDist (const Dist & *dist,* double *ubound*) [inline]

Definition at line 34 of file UpperTruncatedDist.h.

References prior_hessian::UpperTruncatedDist< Dist >::set_ubound().

Here is the call graph for this function:

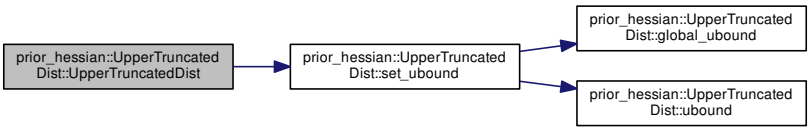


7.66.2.6 template<class Dist> prior_hessian::UpperTruncatedDist< Dist >::UpperTruncatedDist (Dist && *dist,* double *ubound*) [inline]

Definition at line 35 of file UpperTruncatedDist.h.

References prior_hessian::UpperTruncatedDist< Dist >::set_ubound().

Here is the call graph for this function:



7.66.3 Member Function Documentation

7.66.3.1 template<class Dist > double prior_hessian::UpperTruncatedDist< Dist >::cdf (double *x*) const

Definition at line 141 of file UpperTruncatedDist.h.

Referenced by prior_hessian::UpperTruncatedDist< Dist >::median().

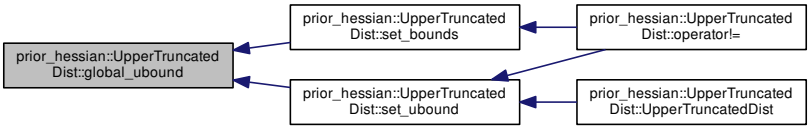
Here is the caller graph for this function:

7.66.3.2 template<class Dist> static double prior_hessian::UpperTruncatedDist< Dist >::global_ubound ( ) [inline], [static]

Definition at line 38 of file UpperTruncatedDist.h.

Referenced by prior_hessian::UpperTruncatedDist< Dist >::set_bounds(), and prior_hessian::UpperTruncatedDist< Dist >::set_ubound().

Here is the caller graph for this function:

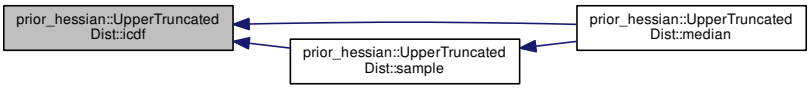


7.66.3.3 template<class Dist > double prior_hessian::UpperTruncatedDist< Dist >::icdf (double *u*) const

Definition at line 147 of file UpperTruncatedDist.h.

Referenced by prior_hessian::UpperTruncatedDist< Dist >::median(), and prior_hessian::UpperTruncatedDist< Dist >::sample().

Here is the caller graph for this function:



7.66.3.4 template<class Dist > double prior_hessian::UpperTruncatedDist< Dist >::llh (double *x*) const

Definition at line 159 of file UpperTruncatedDist.h.

Referenced by prior_hessian::UpperTruncatedDist< Dist >::median().

Here is the caller graph for this function:

7.66.3.5 template<class Dist> double prior_hessian::UpperTruncatedDist< Dist >::mean ( ) const [inline]

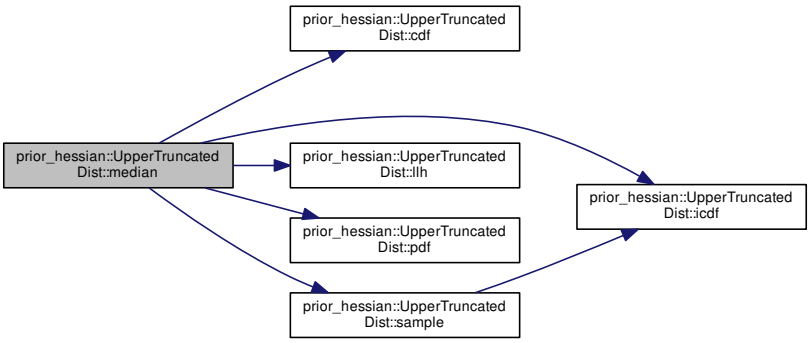
Definition at line 52 of file UpperTruncatedDist.h.

7.66.3.6 template<class Dist> double prior_hessian::UpperTruncatedDist< Dist >::median ( ) const [inline]

Definition at line 53 of file UpperTruncatedDist.h.

References prior_hessian::UpperTruncatedDist< Dist >::cdf(), prior_hessian::UpperTruncatedDist< Dist >::icdf(), prior_hessian::UpperTruncatedDist< Dist >::llh(), prior_hessian::UpperTruncatedDist< Dist >::pdf(), and prior_↩ hessian::UpperTruncatedDist< Dist >::sample().

Here is the call graph for this function:

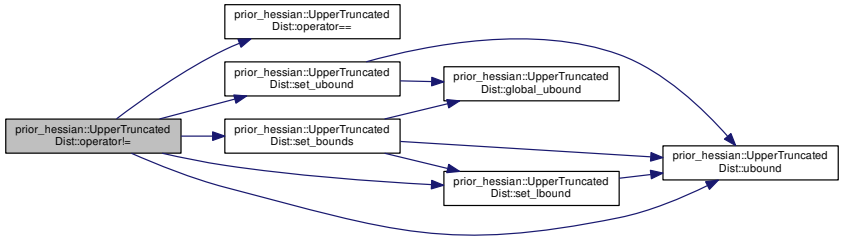


7.66.3.7 template<class Dist> bool prior_hessian::UpperTruncatedDist< Dist >::operator!= (const UpperTruncatedDist< Dist > & *o*) const [inline]

Definition at line 46 of file UpperTruncatedDist.h.

References prior_hessian::UpperTruncatedDist< Dist >::operator==(), prior_hessian::UpperTruncatedDist< Dist >::set_bounds(), prior_hessian::UpperTruncatedDist< Dist >::set_lbound(), prior_hessian::UpperTruncatedDist< Dist >::set_ubound(), and prior_hessian::UpperTruncatedDist< Dist >::ubound().

Here is the call graph for this function:



7.66.3.8 template<class Dist> bool prior_hessian::UpperTruncatedDist< Dist >::operator== (const UpperTruncatedDist< Dist > & *o*) const [inline]

Definition at line 40 of file UpperTruncatedDist.h.

Referenced by prior_hessian::UpperTruncatedDist< Dist >::operator!=().

Here is the caller graph for this function:

7.66.3.9 template<class Dist > double prior_hessian::UpperTruncatedDist< Dist >::pdf (double *x*) const

Definition at line 153 of file UpperTruncatedDist.h.

Referenced by prior_hessian::UpperTruncatedDist< Dist >::median().

Here is the caller graph for this function:

7.66.3.10 template<class Dist > template<class RngT > double prior_hessian::UpperTruncatedDist< Dist >::sample (RngT & *rng*) const

Definition at line 166 of file UpperTruncatedDist.h.

References prior_hessian::UpperTruncatedDist< Dist >::icdf().

Referenced by prior_hessian::UpperTruncatedDist< Dist >::median().

Here is the call graph for this function:

Here is the caller graph for this function:

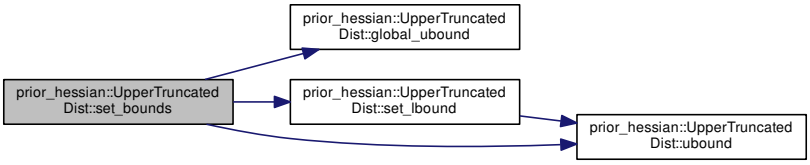
7.66.3.11 template<class Dist > void prior_hessian::UpperTruncatedDist< Dist >::set_bounds (double *lbound,* double *ubound*)

Definition at line 73 of file UpperTruncatedDist.h.

References prior_hessian::UpperTruncatedDist< Dist >::global_ubound(), prior_hessian::UpperTruncatedDist< Dist >::set_lbound(), and prior_hessian::UpperTruncatedDist< Dist >::ubound().

Referenced by prior_hessian::UpperTruncatedDist< Dist >::operator!=().

Here is the call graph for this function:



Here is the caller graph for this function:

7.66.3.12 template<class Dist > void prior_hessian::UpperTruncatedDist< Dist >::set_lbound (double *ubound*)

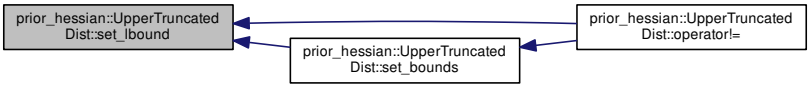
Definition at line 95 of file UpperTruncatedDist.h.

References prior_hessian::UpperTruncatedDist< Dist >::ubound().

Referenced by prior_hessian::UpperTruncatedDist< Dist >::operator!=(), and prior_hessian::UpperTruncatedDist< Dist >::set_bounds().

Here is the call graph for this function:

Here is the caller graph for this function:



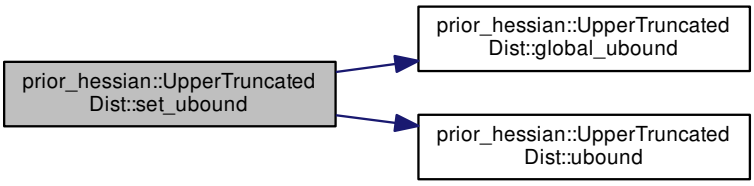
7.66.3.13 template<class Dist > void prior_hessian::UpperTruncatedDist< Dist >::set_ubound (double *ubound*)

Definition at line 111 of file UpperTruncatedDist.h.

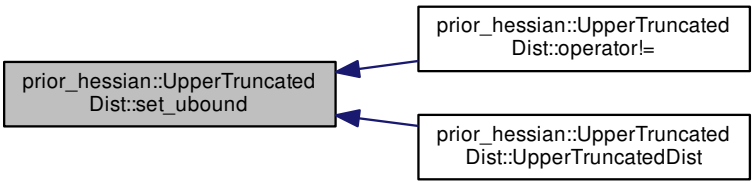
References prior_hessian::UpperTruncatedDist< Dist >::global_ubound(), and prior_hessian::UpperTruncatedDist< Dist >::ubound().

Referenced by prior_hessian::UpperTruncatedDist< Dist >::operator!=(), and prior_hessian::UpperTruncatedDist< Dist >::UpperTruncatedDist().

Here is the call graph for this function:



Here is the caller graph for this function:



7.66.3.14 template<class Dist> bool prior_hessian::UpperTruncatedDist< Dist >::truncated ( ) const [inline]

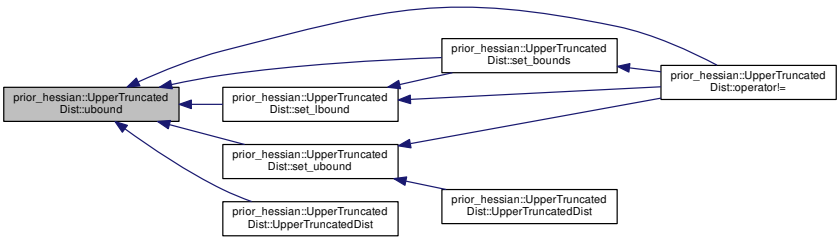
Definition at line 39 of file UpperTruncatedDist.h.

7.66.3.15 template<class Dist> double prior_hessian::UpperTruncatedDist< Dist >::ubound ( ) const [inline]

Definition at line 37 of file UpperTruncatedDist.h.

Referenced by prior_hessian::UpperTruncatedDist< Dist >::operator!=(), prior_hessian::UpperTruncatedDist< Dist >::set_bounds(), prior_hessian::UpperTruncatedDist< Dist >::set_lbound(), prior_hessian::UpperTruncatedDist< Dist >::set_ubound(), and prior_hessian::UpperTruncatedDist< Dist >::UpperTruncatedDist().

Here is the caller graph for this function:



The documentation for this class was generated from the following file:

- UpperTruncatedDist.h

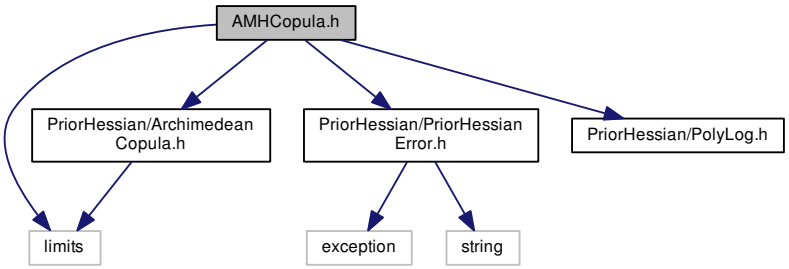
8 File Documentation

8.1 AMHCopula.h File Reference

Ali-Mikhail-Haq Archimedean Copula.

```
#include <limits>
#include "PriorHessian/PriorHessianError.h"
#include "PriorHessian/ArchimedeanCopula.h"
#include "PriorHessian/PolyLog.h"
```

Include dependency graph for AMHCopula.h:



Classes

- class prior_hessian::AMHCopula< Ndim >

Namespaces

- prior_hessian

8.1.1 Detailed Description

Ali-Mikhail-Haq Archimedean Copula.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

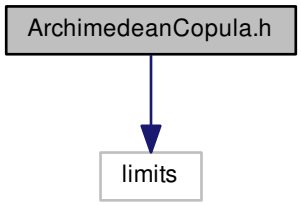
2017-2019

8.2 ArchimedeanCopula.h File Reference

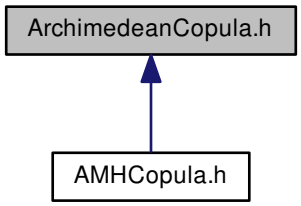
CopulaDist base class.

```
#include <limits>
```

Include dependency graph for ArchimedeanCopula.h:



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



Classes

- class prior_hessian::ArchimedeanCopula
- struct prior_hessian::ArchimedeanCopula::D_GenTerms
- struct prior_hessian::ArchimedeanCopula::D2_GenTerms
- struct prior_hessian::ArchimedeanCopula::D_IGenTerms
- struct prior_hessian::ArchimedeanCopula::D2_IGenTerms
- struct prior_hessian::ArchimedeanCopula::DTheta_GenTerms
- struct prior_hessian::ArchimedeanCopula::D2Theta_GenTerms
- struct prior_hessian::ArchimedeanCopula::DTheta_IGenTerms
- struct prior_hessian::ArchimedeanCopula::D2Theta_IGenTerms

Namespaces

- prior_hessian

8.2.1 Detailed Description

CopulaDist base class.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

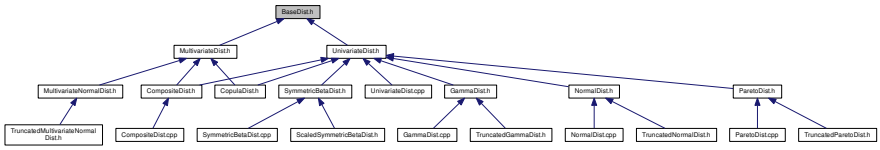
Date

2017 - 2018

8.3 BaseDist.h File Reference

The Base class for UnivariateDist and MultivariateDist.

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



Classes

- class prior_hessian::BaseDist

Namespaces

- prior_hessian

8.3.1 Detailed Description

The Base class for UnivariateDist and MultivariateDist.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

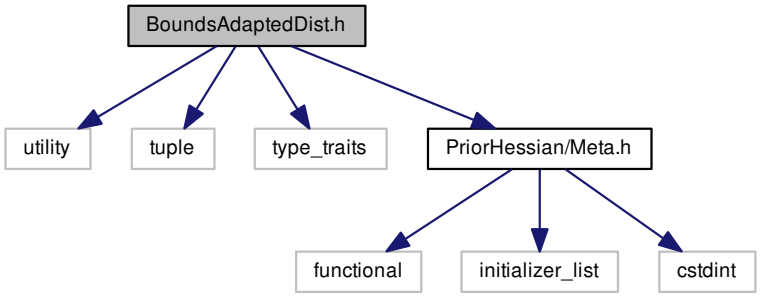
2017-2019

8.4 BoundsAdaptedDist.h File Reference

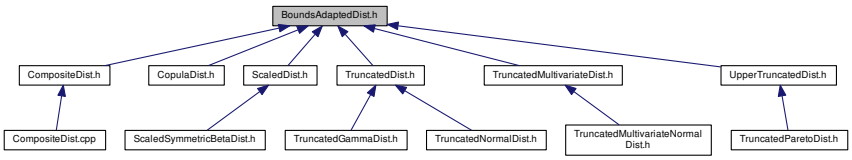
Functions and type-traits to enable easy wrapping of distributions in bounds-modifiable adapters.

```
#include <utility>
#include <tuple>
#include <type_traits>
#include "PriorHessian/Meta.h"
```

Include dependency graph for BoundsAdaptedDist.h:



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



Classes

- class prior_hessian::detail::dist_adaptor_traits< Dist >
- struct prior_hessian::meta::all_dists_are_bounded< DistTs >

Namespaces

- prior_hessian
- prior_hessian::detail
- prior_hessian::meta

Class templates to utilize sequencing behavior of std::initializer_list expressions.

Typedefs

- template<class DistT >
  using prior_hessian::detail::DistTraitsT = detail::dist_adaptor_traits< std::decay_t< DistT >>
- template<class DistT >
  using prior_hessian::BoundsAdaptedDistT = typename detail::dist_adaptor_traits< std::decay_t< DistT >>↩
  ::bounds_adapted_dist
- template<class... DistTs>
  using prior_hessian::meta::ConstructableIfAllDistsAreBoundedT = std::enable_if< all_dists_are_bounded< DistTs... >::value, bool >
- template<class... DistTs>
  using prior_hessian::meta::ConstructableIfNotAllDistsAreBoundedT = std::enable_if<!all_dists_are_bounded< DistTs... >::value, bool >

Functions

- template<class... Ts, std::size_t... I>
  std::tuple< BoundsAdaptedDistT< Ts >... > prior_hessian::detail::make_adapted_bounded_dist_tuple (std↩
  ::tuple< Ts... > &&dists, std::index_sequence< I... >)
- template<class... Ts, std::size_t... I>
  std::tuple< BoundsAdaptedDistT< Ts >... > prior_hessian::detail::make_adapted_bounded_dist_tuple (const std::tuple< Ts... > &dists, std::index_sequence< I... >)
- template<class Dist , typename = meta::EnableIfIsNotTupleT<Dist>>
  std::enable_if_t< detail::DistTraitsT< Dist >::adaptable_bounds, Dist > prior_hessian::make_adapted_↩
  bounded_dist (Dist &&dist)
- template<class Dist , typename = meta::EnableIfIsNotTupleT<Dist>>
  std::enable_if_t< !detail::DistTraitsT< Dist >::adaptable_bounds, BoundsAdaptedDistT< Dist > > prior_↩
  hessian::make_adapted_bounded_dist (Dist &&dist)
- template<class Dist , class Vec , typename = meta::EnableIfIsNotTupleT<Dist>>
  std::enable_if_t< detail::DistTraitsT< Dist >::adaptable_bounds, Dist > prior_hessian::make_adapted_↩
  bounded_dist (Dist &&dist, Vec &&lbound, Vec &&ubound)
- template<class Dist , class Vec , typename = meta::EnableIfIsNotTupleT<Dist>>
  std::enable_if_t<!detail::DistTraitsT< Dist >::adaptable_bounds, BoundsAdaptedDistT< Dist > > prior_↩
  hessian::make_adapted_bounded_dist (Dist &&dist, Vec &&lbound, Vec &&ubound)
- template<class... Ts>
  std::tuple< BoundsAdaptedDistT< Ts >... > prior_hessian::make_adapted_bounded_dist_tuple (Ts &&...ts)
- template<class... Ts>
  std::tuple< BoundsAdaptedDistT< Ts >... > prior_hessian::make_adapted_bounded_dist_tuple (std::tuple< Ts... > &&dists)
- template<class... Ts>
  std::tuple< BoundsAdaptedDistT< Ts >... > prior_hessian::make_adapted_bounded_dist_tuple (const std↩
  ::tuple< Ts... > &dists)

8.4.1 Detailed Description

Functions and type-traits to enable easy wrapping of distributions in bounds-modifiable adapters.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2018-2019 The bounds-adapted version of a distribution is a distribution that has been wrapped by an adapter-class that modifies the distribution so that the bounds can be set to finite values.

The main types of adapters are

- TruncatedDist:
  - Has global_ubound and global_lbound which may both be infinite (e.g., with the NormalDist)
  - Adapts distributions: NormalDist, GammaDist
- UpperTrunctedDist: Adapts distributions which already have an inherit lower-bound as a parameter
  - Adapts distributions: ParatoDist
- ScaledDist:
  - Adapts distributions with finite domain like the Beta distribution by scaling them to arbitrary finite bounds
  - Adapts distributions: BetaDist, SymmetricBetaDist

Functions: make_adapted_bounded_dist

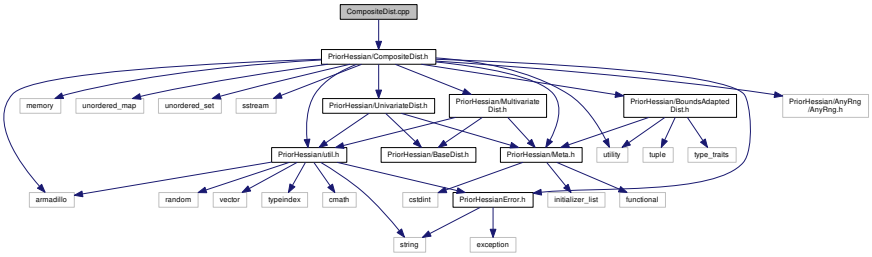
- Make a bounds-adapted version of given distribution make_adapted_bounded_dist_tuple -
- Make a tuple of bounds-adapted version of given distributions

8.5 CompositeDist.cpp File Reference

CompositeDist and associated classes and nested classes.

```
#include "PriorHessian/CompositeDist.h"
```

Include dependency graph for CompositeDist.cpp:



Namespaces

- prior_hessian

Functions

- std::ostream & prior_hessian::operator<< (std::ostream &out, const CompositeDist &comp_dist)

8.5.1 Detailed Description

CompositeDist and associated classes and nested classes.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

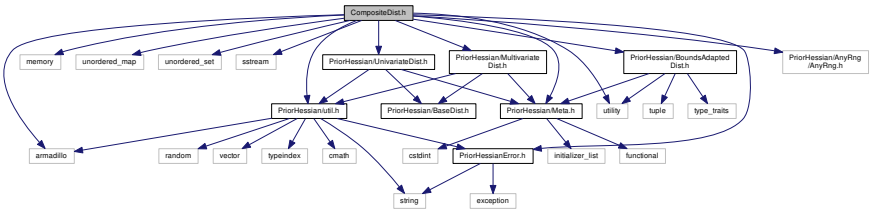
2017-2019

8.6 CompositeDist.h File Reference

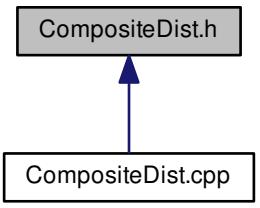
CompositeDist class declaration and inline and templated member function definitions.

```
#include <utility>
#include <memory>
#include <unordered_map>
#include <unordered_set>
#include <sstream>
#include <armadillo>
#include "PriorHessian/Meta.h"
#include "PriorHessian/util.h"
#include "PriorHessian/PriorHessianError.h"
#include "PriorHessian/UnivariateDist.h"
#include "PriorHessian/MultivariateDist.h"
#include "PriorHessian/BoundsAdaptedDist.h"
#include "PriorHessian/AnyRng/AnyRng.h"
```

Include dependency graph for CompositeDist.h:



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



Classes

- class prior_hessian::CompositeDist

  A probability distribution made of independent component distributions composing groups of 1 or more variables.
- class prior_hessian::CompositeDist::ComponentDistAdaptor< DistT, Enable >
- class prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, Univariate↩ Dist > >
- class prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, Multivariate↩ Dist > >

Namespaces

- prior_hessian

Functions

- std::ostream & prior_hessian::operator<< (std::ostream &out, const CompositeDist &comp_dist)

8.6.1 Detailed Description

CompositeDist class declaration and inline and templated member function definitions.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017-2019

8.6.2 Class Documentation

8.6.2.1 class prior_hessian::CompositeDist::ComponentDistAdaptor

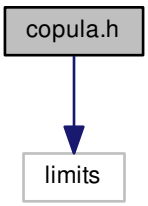
template<class DistT, typename Enable = void>
class prior_hessian::CompositeDist::ComponentDistAdaptor< DistT, Enable >

Definition at line 64 of file CompositeDist.h.

8.7 copula.h File Reference

CopulaDist base class.

`#include <limits>`
Include dependency graph for copula.h:



8.7.1 Detailed Description

CopulaDist base class.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

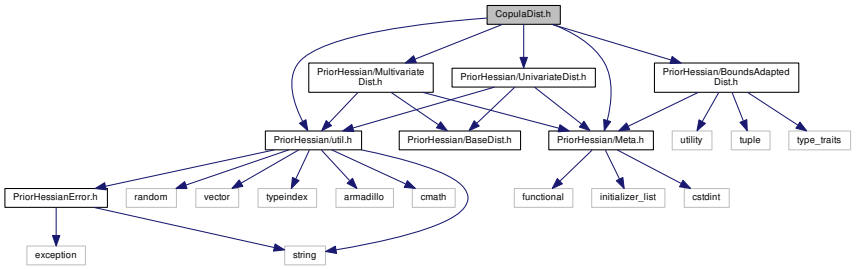
2017-2019

8.8 CopulaDist.h File Reference

CopulaDist base class.

```
#include "PriorHessian/util.h"
#include "PriorHessian/Meta.h"
#include "PriorHessian/UnivariateDist.h"
#include "PriorHessian/MultivariateDist.h"
#include "PriorHessian/BoundsAdaptedDist.h"
```

Include dependency graph for CopulaDist.h:



Classes

- class prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >
- struct prior_hessian::detail::dist_adaptor_traits< CopulaDistImpl::CopulaDist< CopulaTemplate, DistTs... > >

Namespaces

- prior_hessian
- prior_hessian::CopulaDistImpl
- prior_hessian::detail

Typedefs

- template<template< int > class CopulaTemplate, class... MarginalDistTs>
  using prior_hessian::CopulaDist = CopulaDistImpl::CopulaDist< CopulaTemplate, BoundsAdaptedDistT< MarginalDistTs >... >

Functions

- template<template< int > class CopulaTemplate, class... MarginalDistTs>
  CopulaDist< CopulaTemplate, MarginalDistTs... > prior_hessian::make_copula_dist (CopulaTemplate< sizeof...(MarginalDistTs)> &&copula, MarginalDistTs &&...dists)

8.8.1 Detailed Description

CopulaDist base class.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

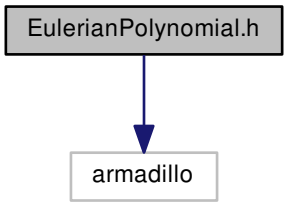
2017-2019

8.9 EulerianPolynomial.h File Reference

EulerianPolynomial computation.

```
#include <armadillo>
```

Include dependency graph for EulerianPolynomial.h:



Classes

- struct prior_hessian::eulerian_number< N, M >
- struct prior_hessian::eulerian_number< 0, M >

Namespaces

- prior_hessian
- prior_hessian::detail

Functions

- template<long N, long... I>
  VecT prior_hessian::detail::eulerian_polynomial ()
- template<long N>
  VecT prior_hessian::eulerian_polynomial ()

8.9.1 Detailed Description

EulerianPolynomial computation.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

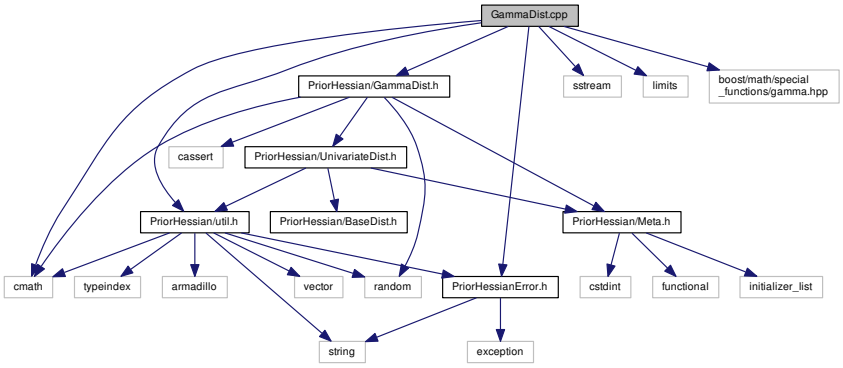
2017-2019

8.10 GammaDist.cpp File Reference

GammaDist class definition.

```
#include "PriorHessian/GammaDist.h"
#include "PriorHessian/util.h"
#include "PriorHessian/PriorHessianError.h"
#include <cmath>
#include <sstream>
#include <limits>
#include <boost/math/special_functions/gamma.hpp>
```

Include dependency graph for GammaDist.cpp:



Namespaces

- prior_hessian

8.10.1 Detailed Description

GammaDist class definition.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

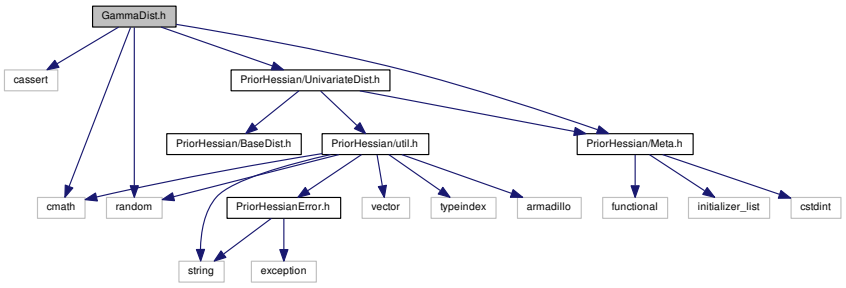
2017-2019

8.11 GammaDist.h File Reference

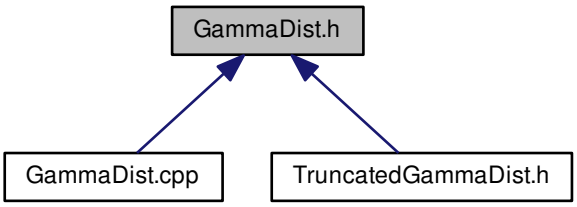
GammaDist class declaration and templated methods.

```
#include <cassert>
#include <cmath>
#include <random>
#include "PriorHessian/Meta.h"
#include "PriorHessian/UnivariateDist.h"
```

Include dependency graph for GammaDist.h:



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



Classes

- class prior_hessian::GammaDist

  Gamma distribution.

Namespaces

- prior_hessian

8.11.1 Detailed Description

GammaDist class declaration and templated methods.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

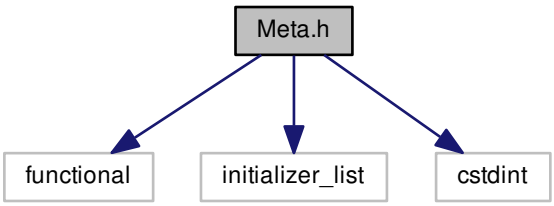
2017-2019

8.12 Meta.h File Reference

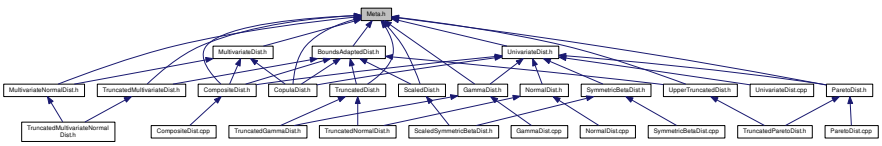
Enables the use of variadic templates in interesting ways.

```
#include <functional>
#include <initializer_list>
#include <cstdint>
```

Include dependency graph for Meta.h:



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



Classes

- struct prior_hessian::meta::conjunction<... >
- struct prior_hessian::meta::conjunction< B1 >
- struct prior_hessian::meta::conjunction< B1, Bn... >
- struct prior_hessian::meta::disjunction<... >
- struct prior_hessian::meta::disjunction< B1 >
- struct prior_hessian::meta::disjunction< B1, Bn... >
- struct prior_hessian::meta::is_template_of< class, typename >
- struct prior_hessian::meta::is_template_of< ClassTemplate, ClassTemplate< Ts... > >
- struct prior_hessian::meta::is_numeric_template_of< class, typename >
- struct prior_hessian::meta::is_numeric_template_of< ClassNumericTemplate, ClassNumericTemplate< Is... > >
- class prior_hessian::meta::is_subclass_of_numeric_template< T, U >
- class prior_hessian::meta::is_copula< T, U >

Namespaces

- prior_hessian
- prior_hessian::meta

  Class templates to utilize sequencing behavior of std::initializer_list expressions.

Macros

- #define GCC_VERSION (__GNUC__ * 100 + __GNUC_MINOR__ * 10 + __GNUC_PATCHLEVEL__)
- #define PRIOR_HESSIAN_META_CONSTEXPR constexpr
- #define PRIOR_HESSIAN_META_HAS_CONSTEXPR 1

Typedefs

- template<template< template< int > class, class... > class CopulaT, class U >
  using prior_hessian::meta::ConstructableIfIsCopulaT = std::enable_if_t< is_copula< CopulaT, U >::value,bool >
- template<class ReturnT , class BoolT >
  using prior_hessian::meta::ReturnIfT = std::enable_if_t< BoolT::value, ReturnT >
- template<bool val>
  using prior_hessian::meta::ConstructableIf = std::enable_if_t< val, bool >
- template<bool val>
  using prior_hessian::meta::ConstructableIfNot = std::enable_if_t<!val, bool >
- template<class T , class SelfT >
  using prior_hessian::meta::ConstructableIfNotSelfT = std::enable_if_t<!std::is_same< std::decay_t< T >, SelfT >::value, bool >
- template<class T , class BaseT >
  using prior_hessian::meta::EnableIfSubclassT = std::enable_if_t< std::is_base_of< std::remove_reference_t< BaseT >, std::remove_reference_t< T >>::value >
- template<class T , template< int > class ClassNumericTemplate>
  using prior_hessian::meta::EnableIfSubclassOfNumericTemplateT = std::enable_if_t< is_subclass_of_numeric_template< ClassNumericTemplate, std::remove_reference_t< T >>::value >

- template<class ReturnT , class T , template< int > class ClassNumericTemplate>
  using prior_hessian::meta::ReturnIfSubclassOfNumericTemplateT = std::enable_if_t< is_subclass_of_numeric↩
  _template< ClassNumericTemplate, std::remove_reference_t< T >>::value, ReturnT >
- template<class T , class SelfT >
  using prior_hessian::meta::EnableIfNotIsSelfT = std::enable_if_t< !std::is_same< std::decay_t< T >, SelfT >↩
  ::value >
- template<class ReturnT , class T , class BaseT >
  using prior_hessian::meta::ReturnIfSubclassT = std::enable_if_t< std::is_base_of< std::remove_reference_t<
  BaseT >, std::remove_reference_t< T >>::value, ReturnT >
- template<class BaseT , class... Ts>
  using prior_hessian::meta::EnableIfIsSuperclassOfAllT = std::enable_if_t< conjunction< std::is_base_of< std↩
  ::remove_reference_t< BaseT >, std::remove_reference_t< Ts >>... >::value >
- template<class T , template< typename... > class ClassTemplate>
  using prior_hessian::meta::EnableIfInstantiatedFromT = std::enable_if_t< is_template_of< ClassTemplate, std↩
  ::remove_reference_t< T >>::value >
- template<class T , template< int > class ClassTemplate>
  using prior_hessian::meta::EnableIfInstantiatedFromNumericT = std::enable_if_t< is_numeric_template_of<
  ClassTemplate, std::remove_reference_t< T >>::value >
- template<class ReturnT , class T , template< int > class ClassTemplate>
  using prior_hessian::meta::ReturnIfInstantiatedFromNumericT = std::enable_if_t< is_numeric_template_of<
  ClassTemplate, std::remove_reference_t< T >>::value, ReturnT >
- template<class T , template< typename... > class ClassTemplate>
  using prior_hessian::meta::EnableIfNotInstantiatedFromT = std::enable_if_t< !is_template_of< ClassTemplate,
  std::remove_reference_t< T >>::value >
- template<class ReturnT , class TestT , template< typename... > class ClassTemplate>
  using prior_hessian::meta::ReturnIfInstantiatedFromT = std::enable_if_t< is_template_of< ClassTemplate, std↩
  ::remove_reference_t< TestT >>::value, ReturnT >
- template<class ReturnT , class TestT , template< typename... > class ClassTemplate>
  using prior_hessian::meta::ReturnIfNotInstantiatedFromT = std::enable_if_t< !is_template_of< ClassTemplate,
  std::remove_reference_t< TestT >>::value, ReturnT >
- template<template< typename > class ClassTemplate, class... Ts>
  using prior_hessian::meta::EnableIfIsTemplateForAllT = std::enable_if_t< conjunction< is_template_of<
  ClassTemplate, std::remove_reference_t< Ts >>... >::value >
- template<template< typename... > class ClassTemplate, class... Ts>
  using prior_hessian::meta::ConstructableIfIsTemplateForAllT = std::enable_if_t< conjunction< is_template_of<
  ClassTemplate, std::remove_reference_t< Ts >>... >::value, bool >
- template<class SuperClass , class T >
  using prior_hessian::meta::ConstructableIfIsSuperClassT = std::enable_if_t< std::is_base_of< std::remove_↩
  reference_t< SuperClass >, std::remove_reference_t< T >>::value, bool >
- template<class SuperClass , class... Ts>
  using prior_hessian::meta::ConstructableIfIsSuperClassForAllT = std::enable_if_t< conjunction< std::is_base↩
  _of< std::remove_reference_t< SuperClass >, std::remove_reference_t< Ts >>... >::value, bool >
- template<class T , template< int > class ClassTemplate>
  using prior_hessian::meta::ConstructableIfInstantiatedFromNumericT = std::enable_if_t< is_numeric_template↩
  _of< ClassTemplate, std::remove_reference_t< T >>::value, bool >
- template<class T >
  using prior_hessian::meta::EnableIfIsNotTupleT = std::enable_if_t< !is_template_of< std::tuple, std::remove_↩
  reference_t< T >>::value >
- template<class... Ts>
  using prior_hessian::meta::EnableIfNonEmpty = std::enable_if_t< (sizeof...(Ts)>0) >
- template<class... Ts>
  using prior_hessian::meta::EnableIfAllAreNotTupleT = std::enable_if_t< !disjunction< is_template_of< std::tuple,
  std::remove_reference_t< Ts >>... >::value >

- template<class SelfT , class T >
  using prior_hessian::meta::EnableIfIsNotTupleAndIsNotSelfT = std::enable_if_t< !is_template_of< std::tuple, std::remove_reference_t< T >>::value &&!std::is_same< std::decay_t< T >, SelfT >::value >
- template<class T , class... Ts>
  using prior_hessian::meta::ConstructableIfAllAreNotTupleAndAreNotT = std::enable_if_t< !disjunction< is_↩ template_of< std::tuple, std::remove_reference_t< Ts >>... >::value &&!disjunction< std::is_same< std↩ ::decay_t< Ts >, T >... >::value, bool >
- template<class Dist , class BaseDist >
  using prior_hessian::meta::DerivedFrom = std::enable_if_t< std::is_base_of< std::decay_t< BaseDist >, std↩ ::decay_t< Dist >>::value, std::decay_t< Dist >>

Functions

- template<class T = int>
  void prior_hessian::meta::call_in_order (std::initializer_list< T >)
- template<class InputIterator , class ResultT , class BinaryOperation >
  PRIOR_HESSIAN_META_CONSTEXPR ResultT prior_hessian::meta::constexpr_accumulate (InputIterator first, InputIterator last, ResultT init, BinaryOperation op)
- PRIOR_HESSIAN_META_CONSTEXPR bool prior_hessian::meta::logical_and_in_order (std::initializer_list< bool > L)
- template<class T >
  PRIOR_HESSIAN_META_CONSTEXPR T prior_hessian::meta::sum_in_order (std::initializer_list< T > L)
- template<class T >
  PRIOR_HESSIAN_META_CONSTEXPR T prior_hessian::meta::prod_in_order (std::initializer_list< T > L)

8.12.1 Detailed Description

Enables the use of variadic templates in interesting ways.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017-2019

8.12.2 Macro Definition Documentation

8.12.2.1 #define GCC_VERSION (__GNUC__ * 100 + __GNUC_MINOR__ * 10 + __GNUC_PATCHLEVEL__)

Definition at line 11 of file Meta.h.

8.12.2.2 #define PRIOR_HESSIAN_META_CONSTEXPR constexpr

Definition at line 15 of file Meta.h.

Referenced by prior_hessian::meta::call_in_order(), prior_hessian::meta::constexpr_accumulate(), prior_hessian↩ ::meta::logical_and_in_order(), and prior_hessian::meta::sum_in_order().

8.12.2.3 #define PRIOR_HESSIAN_META_HAS_CONSTEXPR 1

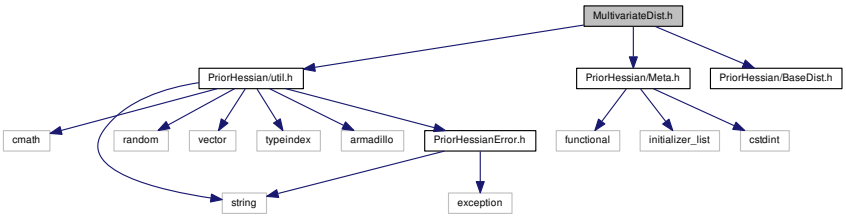
Definition at line 16 of file Meta.h.

8.13 MultivariateDist.h File Reference

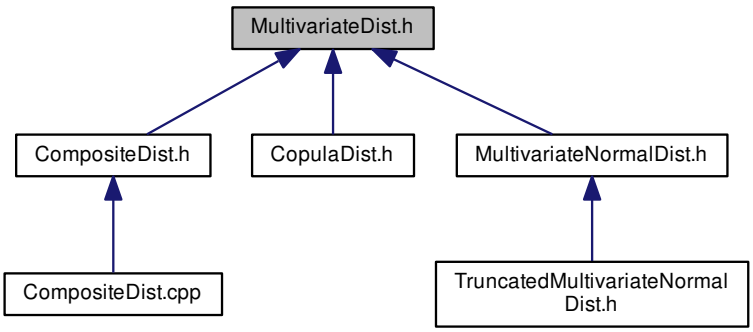
MultivariateDist base class.

```
#include "PriorHessian/util.h"
#include "PriorHessian/Meta.h"
#include "PriorHessian/BaseDist.h"
```

Include dependency graph for MultivariateDist.h:



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



Classes

- class prior_hessian::MultivariateDist

Namespaces

- prior_hessian

8.13.1 Detailed Description

MultivariateDist base class.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

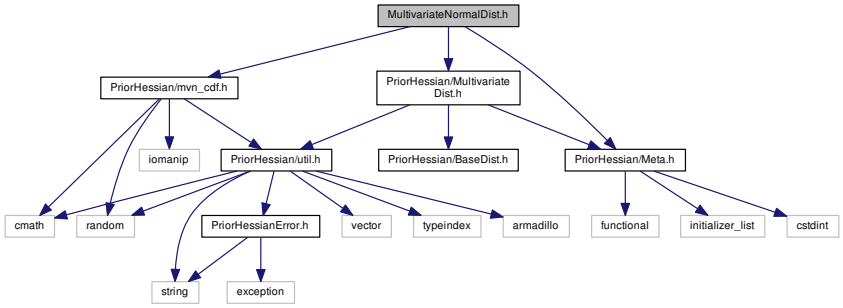
2017-2019

8.14 MultivariateNormalDist.h File Reference

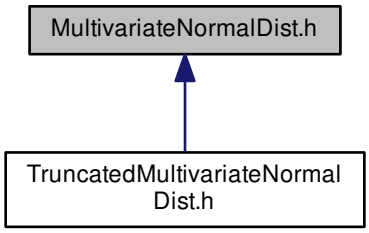
MultivariateNormalDist base class.

```
#include "PriorHessian/MultivariateDist.h"
#include "PriorHessian/Meta.h"
#include "PriorHessian/mvn_cdf.h"
```

Include dependency graph for MultivariateNormalDist.h:



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



Classes

- class prior_hessian::MultivariateNormalDist< Ndim >

  Multivariate Normal distribution.

Namespaces

- prior_hessian
- prior_hessian::helpers

Functions

- template<class Vec , class Mat >
  double prior_hessian::helpers::compute_quadratic_from_symmetric (IdxT Ndim, const Vec &v, const Mat &A)

8.14.1 Detailed Description

MultivariateNormalDist base class.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

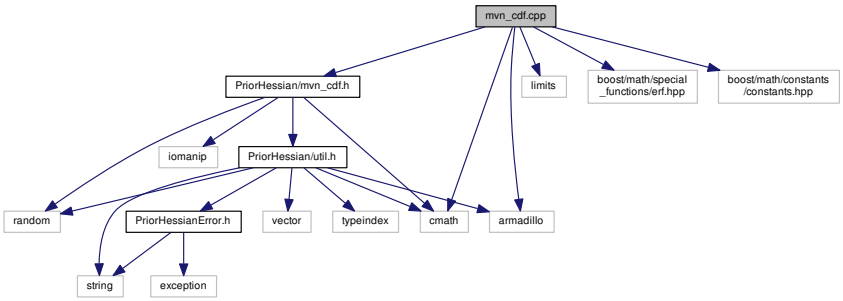
Date

2017-2019

8.15 mvn_cdf.cpp File Reference

```
#include "PriorHessian/mvn_cdf.h"
#include <cmath>
#include <limits>
#include <armadillo>
#include <boost/math/special_functions/erf.hpp>
#include <boost/math/constants/constants.hpp>
```

Include dependency graph for mvn_cdf.cpp:



Namespaces

- prior_hessian

Functions

- double prior_hessian::unit_normal_cdf (double t)
- double prior_hessian::unit_normal_icdf (double u)
- double prior_hessian::bounded (double x)
- double prior_hessian::owen_t_integral (double h, double a, double gh)
- double prior_hessian::owen_b_integral (double h, double k, double r)
- double prior_hessian::donnelly_bvn_integral (double ah, double ak, double r)
- double prior_hessian::donnelly_bvn_integral_orig (double ah, double ak, double r)

8.15.1 Detailed Description

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

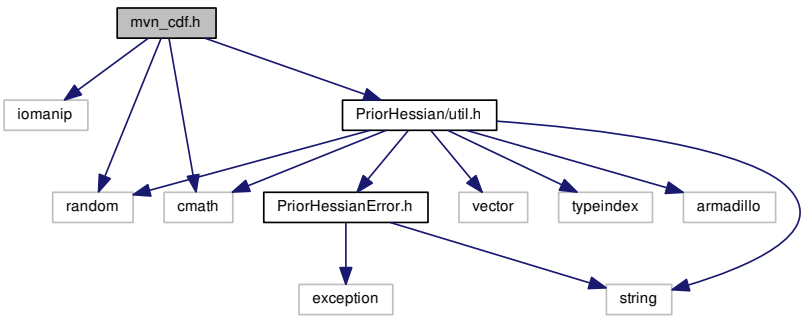
2017-2019

8.16 mvn_cdf.h File Reference

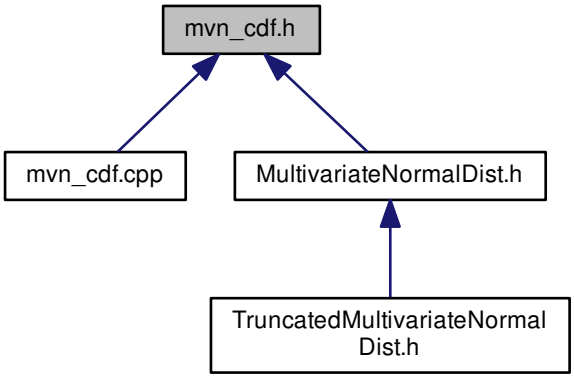
Numerical computation of multivariate normal cdfs in 2,3 and higher dims.

```
#include <iomanip>
#include <random>
#include <cmath>
#include "PriorHessian/util.h"
```

Include dependency graph for mvn_cdf.h:



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



Namespaces

- prior_hessian
- prior_hessian::genz
- prior_hessian::genz::fortran

Functions

- double prior_hessian::unit_normal_cdf (double t)
- double prior_hessian::unit_normal_icdf (double u)
- double prior_hessian::owen_t_integral (double h, double a, double gh)
- double prior_hessian::owen_t_integral (double h, double a)
- double prior_hessian::owen_b_integral (double h, double k, double r)
- double prior_hessian::donnelly_bvn_integral (double ah, double ak, double r)
- double prior_hessian::donnelly_bvn_integral_orig (double ah, double ak, double r)
- template<class Vec , class Mat >
  double prior_hessian::donnelly_bvn_cdf (const Vec &b, const Mat &sigma)
- template<class Vec , class Mat >
  double prior_hessian::owen_bvn_cdf (const Vec &b, const Mat &sigma)
- template<class Vec , class Mat >
  double prior_hessian::mc_mvn_integral (const Vec &a, const Vec &b, const Mat &U, double &error, int &niter)
- template<class Vec , class Mat >
  double prior_hessian::mc_mvn_cdf_core (const Vec &b, const Mat &U, double &error, int &niter)
- template<class Vec , class Mat >
  double prior_hessian::mc_mvn_cdf (const Vec &b, const Mat &S, double &error)
- int prior_hessian::genz::fortran::mvndst_ (int *n, double lower[], double upper[], int infin[], double correl[], int *maxpts, double *abseps, double *releps, double *error, double *value, int *inform)
- template<class Vec , class Mat >
  double prior_hessian::genz::mvn_cdf_genz (const Vec &b, const Mat &S, double &error)

8.16.1 Detailed Description

Numerical computation of multivariate normal cdfs in 2,3 and higher dims.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

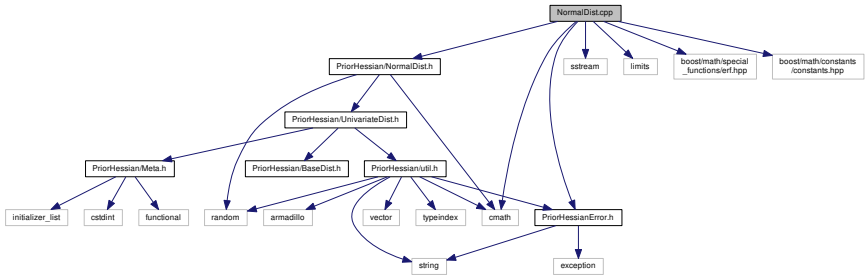
2017-2019

8.17 NormalDist.cpp File Reference

NormalDist class definition.

```
#include "PriorHessian/NormalDist.h"
#include "PriorHessian/PriorHessianError.h"
#include <sstream>
#include <cmath>
#include <limits>
#include <boost/math/special_functions/erf.hpp>
#include <boost/math/constants/constants.hpp>
```

Include dependency graph for NormalDist.cpp:



Namespaces

- prior_hessian

8.17.1 Detailed Description

NormalDist class definition.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

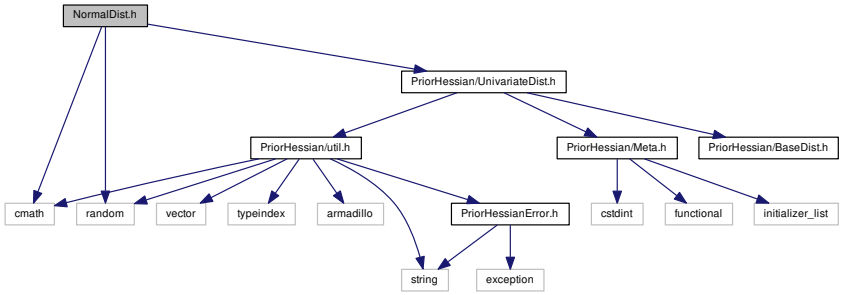
2017-2019

8.18 NormalDist.h File Reference

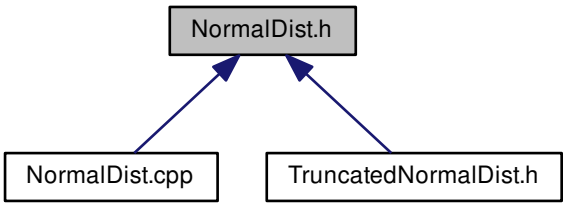
NormalDist class declaration and templated methods.

```
#include <cmath>
#include <random>
#include "PriorHessian/UnivariateDist.h"
```

Include dependency graph for NormalDist.h:



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



Classes

- class prior_hessian::NormalDist

  Normal distribution with truncation.

Namespaces

- prior_hessian

8.18.1 Detailed Description

NormalDist class declaration and templated methods.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

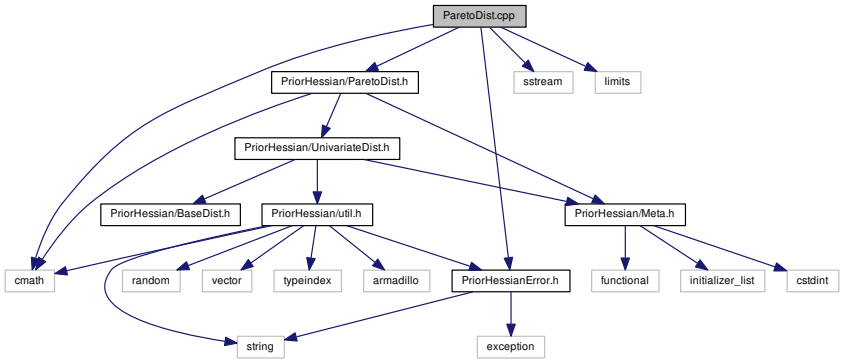
2017-2019

8.19 ParetoDist.cpp File Reference

ParetoDist class definition.

```
#include "PriorHessian/ParetoDist.h"
#include "PriorHessian/PriorHessianError.h"
#include <cmath>
#include <sstream>
#include <limits>
```

Include dependency graph for ParetoDist.cpp:



Namespaces

- prior_hessian

8.19.1 Detailed Description

ParetoDist class definition.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

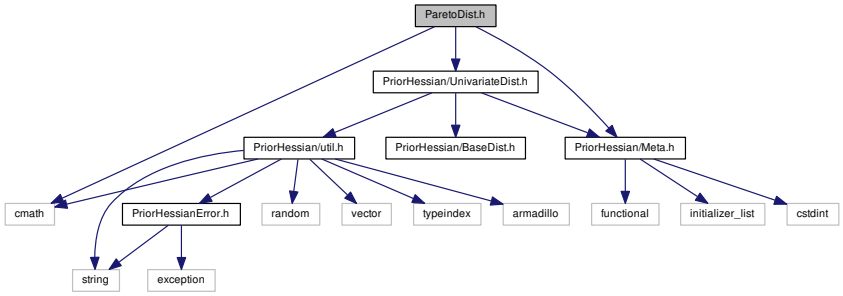
2017-2019

8.20 ParetoDist.h File Reference

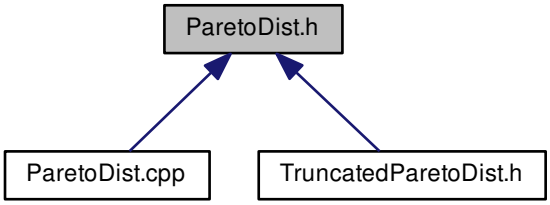
ParetoDist class declaration and templated methods.

```
#include <cmath>
#include "PriorHessian/Meta.h"
#include "PriorHessian/UnivariateDist.h"
```

Include dependency graph for ParetoDist.h:



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



Classes

- class prior_hessian::ParetoDist

  Pareto dist with infinite upper bound.

Namespaces

- prior_hessian

8.20.1 Detailed Description

ParetoDist class declaration and templated methods.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

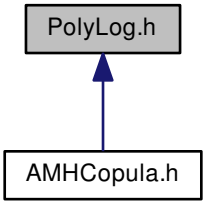
Date

2017-2019

8.21 PolyLog.h File Reference

Poly log computation for negative integer valued parameters.

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



Namespaces

- prior_hessian
- prior_hessian::polylog

Functions

- template<int n>
  double prior_hessian::polylog::eulerian_polynomial (double z)
- template<>
  double prior_hessian::polylog::eulerian_polynomial< 0 > (double)
- template<>
  double prior_hessian::polylog::eulerian_polynomial< 1 > (double z)
- template<>
  double prior_hessian::polylog::eulerian_polynomial< 2 > (double z)

- template<>
  double prior_hessian::polylog::eulerian_polynomial< 3 > (double z)
- template<>
  double prior_hessian::polylog::eulerian_polynomial< 4 > (double z)
- template<>
  double prior_hessian::polylog::eulerian_polynomial< 5 > (double z)
- template<>
  double prior_hessian::polylog::eulerian_polynomial< 6 > (double z)
- template<>
  double prior_hessian::polylog::eulerian_polynomial< 7 > (double z)
- template<>
  double prior_hessian::polylog::eulerian_polynomial< 8 > (double z)
- template<>
  double prior_hessian::polylog::eulerian_polynomial< 9 > (double z)
- template<int n>
  double prior_hessian::polylog::polylog (double z)
- template<>
  double prior_hessian::polylog::polylog< 1 > (double z)

8.21.1 Detailed Description

Poly log computation for negative integer valued parameters.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

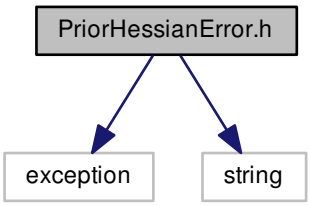
2017-2019

8.22 PriorHessianError.h File Reference

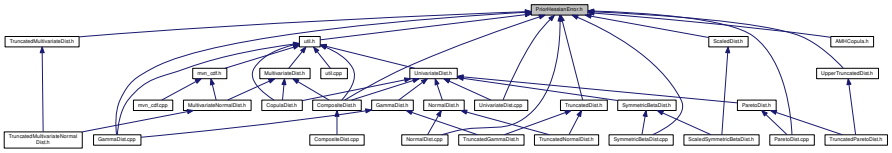
The Exception classes for the PriorHessian library.

```
#include <exception>
#include <string>
```

Include dependency graph for PriorHessianError.h:



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



Classes

- class prior_hessian::PriorHessianError
- struct prior_hessian::IndexError

  Indicates a index access was out of bounds.
- struct prior_hessian::InvalidOperationError
- struct prior_hessian::ParameterSizeError
- struct prior_hessian::ParameterValueError
- struct prior_hessian::ParameterNameError
- struct prior_hessian::ParameterNameUniquenessError
- struct prior_hessian::RuntimeConvergenceError
- struct prior_hessian::RuntimeSamplingError
- struct prior_hessian::RuntimeTypeError
- struct prior_hessian::NumericalOverflowError
- struct prior_hessian::NotImplementedError

Namespaces

- prior_hessian

Macros

- #define DEBUG_ASSERT(...)
- #define ASSERT_SETUP(...)

8.22.1 Detailed Description

The Exception classes for the PriorHessian library.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017-2019

8.22.2 Macro Definition Documentation

8.22.2.1 #define ASSERT_SETUP(...)

Definition at line 37 of file PriorHessianError.h.

8.22.2.2 #define DEBUG_ASSERT(...)

Definition at line 32 of file PriorHessianError.h.

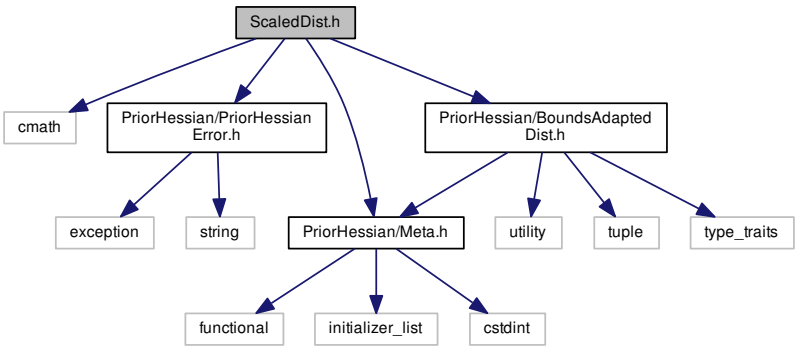
8.23 README.md File Reference

8.24 ScaledDist.h File Reference

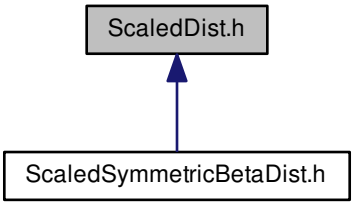
ScaledDist class declaration and templated methods.

```
#include <cmath>
#include "PriorHessian/Meta.h"
#include "PriorHessian/PriorHessianError.h"
#include "PriorHessian/BoundsAdaptedDist.h"
```

Include dependency graph for ScaledDist.h:



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



Classes

- class prior_hessian::ScaledDist< Dist >

Namespaces

- prior_hessian

8.24.1 Detailed Description

ScaledDist class declaration and templated methods.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

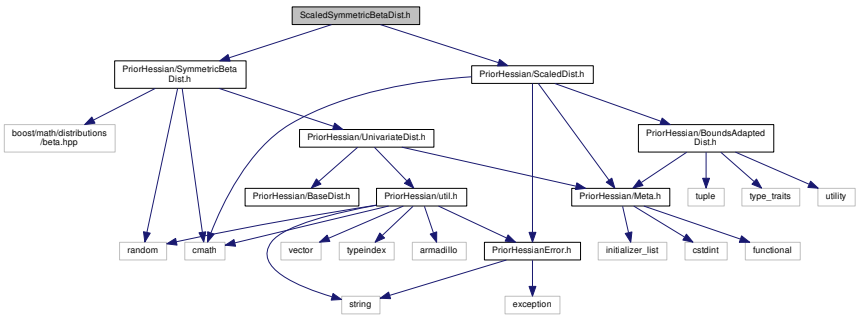
2017-2019

8.25 ScaledSymmetricBetaDist.h File Reference

ScaledSymmetricBetaDist class declaration and templated methods.

```
#include "PriorHessian/SymmetricBetaDist.h"
#include "PriorHessian/ScaledDist.h"
```

Include dependency graph for ScaledSymmetricBetaDist.h:



Classes

- struct prior_hessian::detail::dist_adaptor_traits< SymmetricBetaDist >
- struct prior_hessian::detail::dist_adaptor_traits< ScaledSymmetricBetaDist >

Namespaces

- prior_hessian
- prior_hessian::detail

Typedefs

- using prior_hessian::ScaledSymmetricBetaDist = ScaledDist< SymmetricBetaDist >

Functions

- ScaledSymmetricBetaDist prior_hessian::make_scaled_symmetric_beta_dist (double beta, std::pair< double, double > bounds)

8.25.1 Detailed Description

ScaledSymmetricBetaDist class declaration and templated methods.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

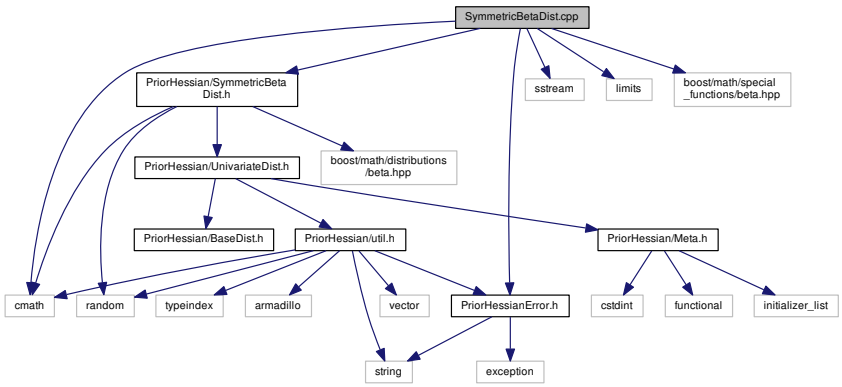
2017-2019

8.26 SymmetricBetaDist.cpp File Reference

SymmetricBetaDist class definition.

```
#include "PriorHessian/SymmetricBetaDist.h"
#include "PriorHessian/PriorHessianError.h"
#include <sstream>
#include <cmath>
#include <limits>
#include <boost/math/special_functions/beta.hpp>
```

Include dependency graph for SymmetricBetaDist.cpp:



Namespaces

- prior_hessian

8.26.1 Detailed Description

SymmetricBetaDist class definition.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

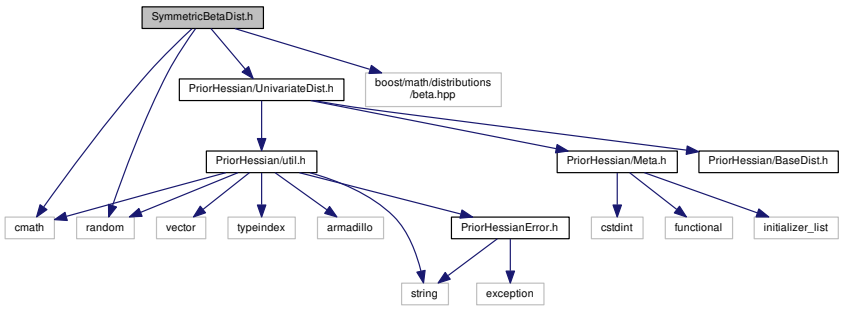
2017-2019

8.27 SymmetricBetaDist.h File Reference

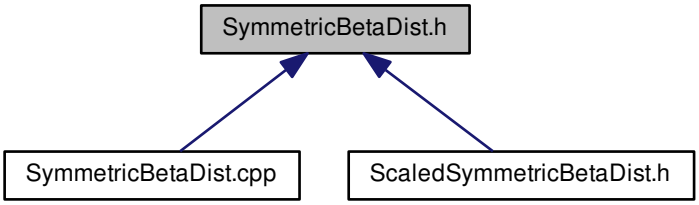
SymmetricBetaDist class declaration and templated methods.

```
#include <cmath>
#include <random>
#include <boost/math/distributions/beta.hpp>
#include "PriorHessian/UnivariateDist.h"
```

Include dependency graph for SymmetricBetaDist.h:



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



Classes

- class prior_hessian::SymmetricBetaDist

  Single parameter beta distribution where $\alpha = \beta$, leading to symmetric bounded distribution.

Namespaces

- prior_hessian

8.27.1 Detailed Description

SymmetricBetaDist class declaration and templated methods.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

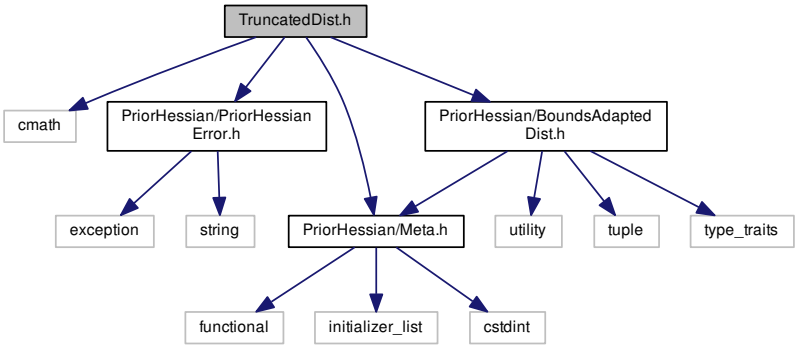
2017-2019

8.28 TruncatedDist.h File Reference

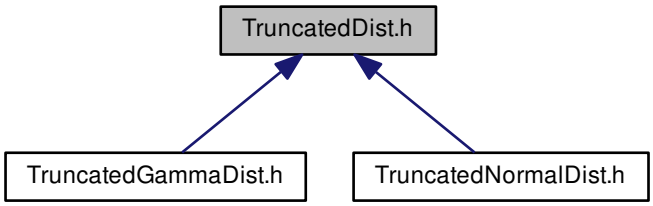
TruncatedDist declaration and templated methods definitions.

```
#include <cmath>
#include "PriorHessian/Meta.h"
#include "PriorHessian/PriorHessianError.h"
#include "PriorHessian/BoundsAdaptedDist.h"
```

Include dependency graph for TruncatedDist.h:



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



Classes

- class prior_hessian::TruncatedDist< Dist >

Namespaces

- prior_hessian

8.28.1 Detailed Description

TruncatedDist declaration and templated methods definitions.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

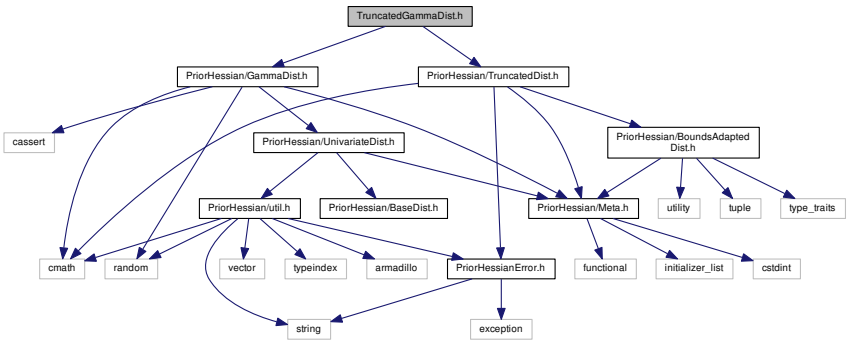
2017-2019

8.29 TruncatedGammaDist.h File Reference

TruncatedGammaDist class declaration and templated methods.

```
#include "PriorHessian/GammaDist.h"
#include "PriorHessian/TruncatedDist.h"
```

Include dependency graph for TruncatedGammaDist.h:



Classes

- struct prior_hessian::detail::dist_adaptor_traits< GammaDist >
- struct prior_hessian::detail::dist_adaptor_traits< TruncatedGammaDist >

Namespaces

- prior_hessian
- prior_hessian::detail

Typedefs

- using prior_hessian::TruncatedGammaDist = TruncatedDist< GammaDist >

Functions

- TruncatedGammaDist prior_hessian::make_bounded_gamma_dist (double scale, double shape, std::pair< double, double > bounds)

8.29.1 Detailed Description

TruncatedGammaDist class declaration and templated methods.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

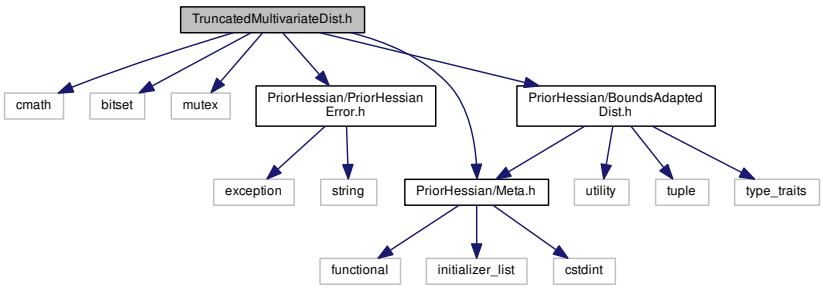
2017-2019

8.30 TruncatedMultivariateDist.h File Reference

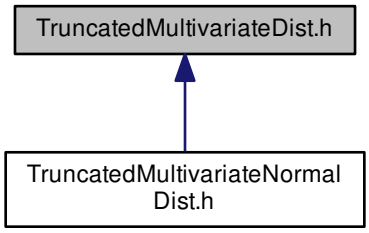
TruncatedMultivariateDist declaration and templated methods definitions.

```
#include <cmath>
#include <bitset>
#include <mutex>
#include "PriorHessian/Meta.h"
#include "PriorHessian/PriorHessianError.h"
#include "PriorHessian/BoundsAdaptedDist.h"
```

Include dependency graph for TruncatedMultivariateDist.h:



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



Classes

- class prior_hessian::mcmc::MCMCData< Ndim >
- class prior_hessian::TruncatedMultivariateDist< Dist >

Namespaces

- prior_hessian
- prior_hessian::mcmc

8.30.1 Detailed Description

TruncatedMultivariateDist declaration and templated methods definitions.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

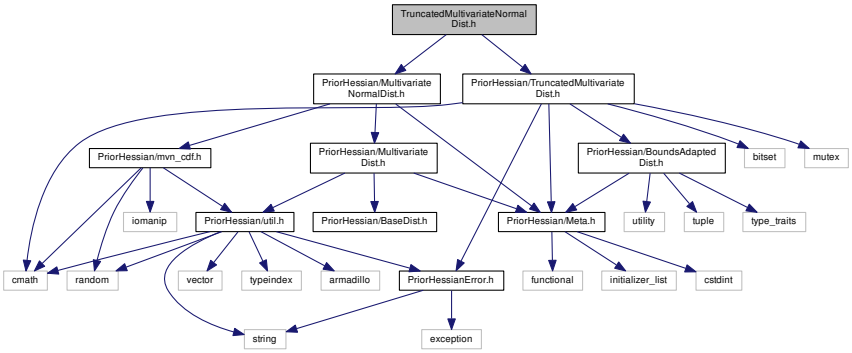
2017-2019

8.31 TruncatedMultivariateNormalDist.h File Reference

TruncatedMultivariateNormalDist class declaration.

```
#include "PriorHessian/MultivariateNormalDist.h"
#include "PriorHessian/TruncatedMultivariateDist.h"
```

Include dependency graph for TruncatedMultivariateNormalDist.h:



Classes

- struct prior_hessian::detail::dist_adaptor_traits< MultivariateNormalDist< Ndim > >
- struct prior_hessian::detail::dist_adaptor_traits< TruncatedMultivariateNormalDist< Ndim > >

Namespaces

- prior_hessian
- prior_hessian::detail

Typedefs

- template<IdxT Ndim>
  using prior_hessian::TruncatedMultivariateNormalDist = TruncatedMultivariateDist< MultivariateNormalDist< Ndim >>

Functions

- template<IdxT Ndim, class Vec , class Mat , class Vec2 >
  TruncatedMultivariateNormalDist< Ndim > prior_hessian::make_bounded_multivariate_normal_dist (Vec &&mu, Mat &&sigma, Vec2 &&lbound, Vec2 &&ubound)

8.31.1 Detailed Description

TruncatedMultivariateNormalDist class declaration.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

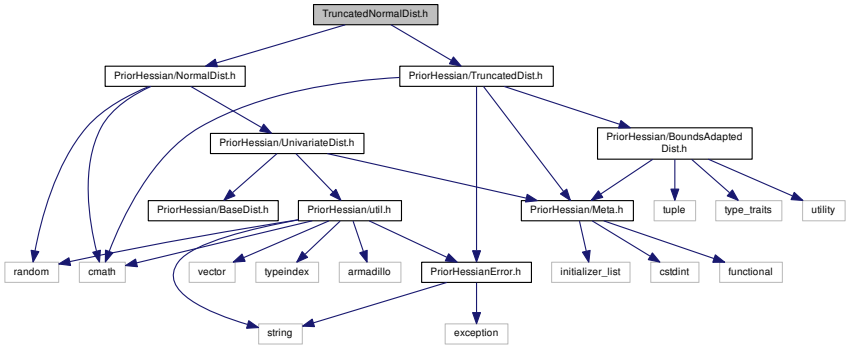
2017-2019

8.32 TruncatedNormalDist.h File Reference

TruncatedNormalDist class declaration.

```
#include "PriorHessian/NormalDist.h"
#include "PriorHessian/TruncatedDist.h"
```

Include dependency graph for TruncatedNormalDist.h:



Classes

- struct prior_hessian::detail::dist_adaptor_traits< NormalDist >
- struct prior_hessian::detail::dist_adaptor_traits< TruncatedNormalDist >

Namespaces

- prior_hessian
- prior_hessian::detail

Typedefs

- using prior_hessian::TruncatedNormalDist = TruncatedDist< NormalDist >

Functions

- TruncatedNormalDist prior_hessian::make_bounded_normal_dist (double mu, double sigma, std::pair< double, double > bounds)

8.32.1 Detailed Description

TruncatedNormalDist class declaration.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

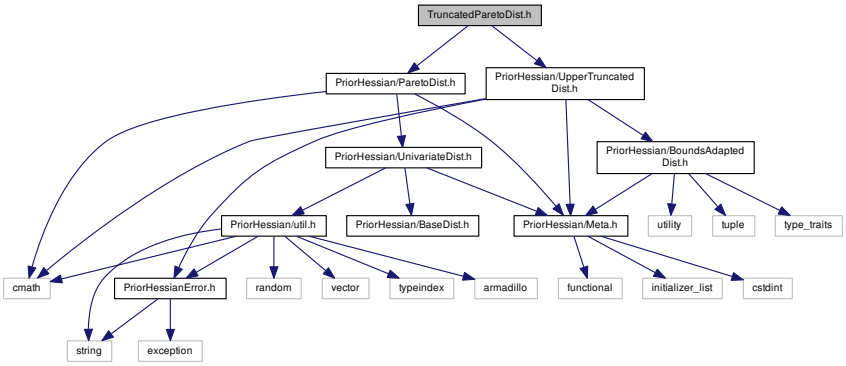
2017-2018

8.33 TruncatedParetoDist.h File Reference

TruncatedParetoDist class declaration and templated methods.

```
#include "PriorHessian/ParetoDist.h"
#include "PriorHessian/UpperTruncatedDist.h"
```

Include dependency graph for TruncatedParetoDist.h:



Classes

- struct prior_hessian::detail::dist_adaptor_traits< ParetoDist >
- struct prior_hessian::detail::dist_adaptor_traits< TruncatedParetoDist >

Namespaces

- prior_hessian
- prior_hessian::detail

Typedefs

- using prior_hessian::TruncatedParetoDist = UpperTruncatedDist< ParetoDist >

Functions

- TruncatedParetoDist prior_hessian::make_bounded_pareto_dist (double alpha, std::pair< double, double > bounds)

8.33.1 Detailed Description

TruncatedParetoDist class declaration and templated methods.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

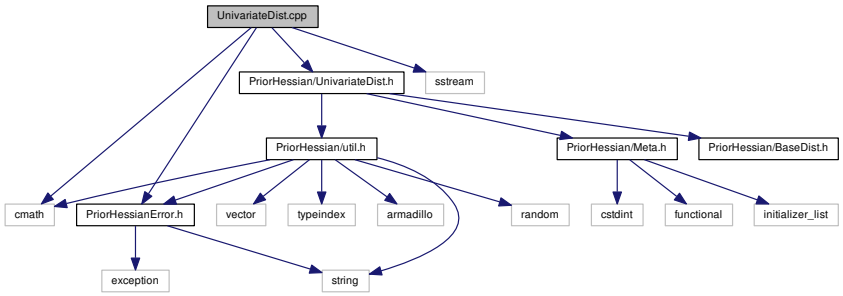
2017-2019

8.34 UnivariateDist.cpp File Reference

UnivariateDist base class method definition.

```
#include "PriorHessian/UnivariateDist.h"
#include "PriorHessian/PriorHessianError.h"
#include <cmath>
#include <sstream>
```

Include dependency graph for UnivariateDist.cpp:



Namespaces

- prior_hessian

8.34.1 Detailed Description

UnivariateDist base class method definition.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

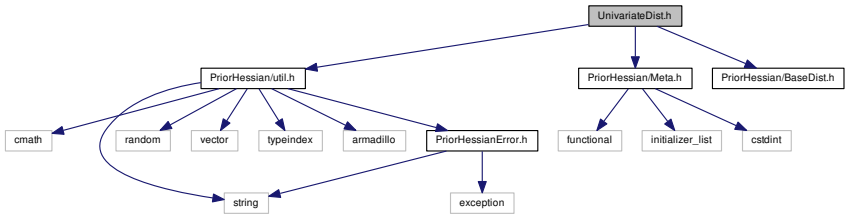
2017-2019

8.35 UnivariateDist.h File Reference

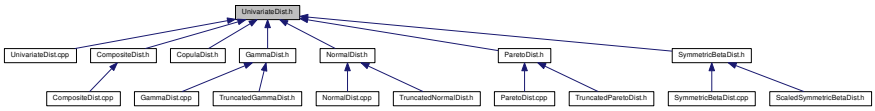
UnivariateDist base class.

```
#include "PriorHessian/util.h"
#include "PriorHessian/Meta.h"
#include "PriorHessian/BaseDist.h"
```

Include dependency graph for UnivariateDist.h:



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



Classes

- class prior_hessian::UnivariateDist

Namespaces

- prior_hessian

8.35.1 Detailed Description

UnivariateDist base class.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

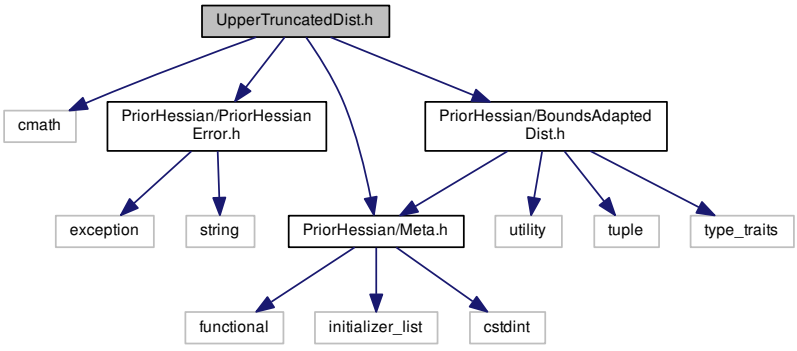
2017-2019

8.36 UpperTruncatedDist.h File Reference

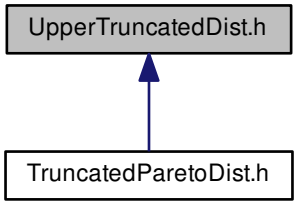
UpperTruncatedDist declaration and templated methods definitions.

```
#include <cmath>
#include "PriorHessian/Meta.h"
#include "PriorHessian/PriorHessianError.h"
#include "PriorHessian/BoundsAdaptedDist.h"
```

Include dependency graph for UpperTruncatedDist.h:



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



Classes

- class prior_hessian::UpperTruncatedDist< Dist >

Namespaces

- prior_hessian

8.36.1 Detailed Description

UpperTruncatedDist declaration and templated methods definitions.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

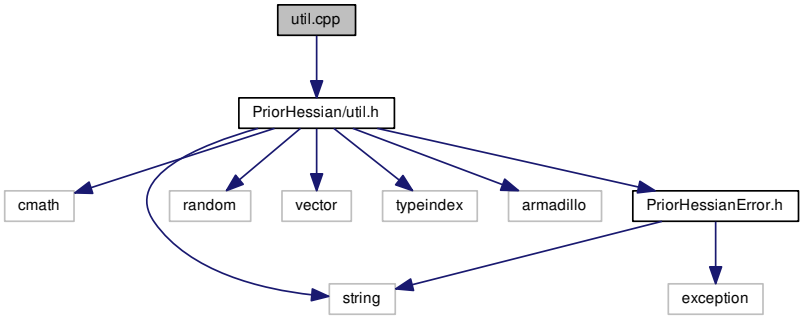
Date

2017-2019

8.37 util.cpp File Reference

Utilities.

```
#include "PriorHessian/util.h"
```
Include dependency graph for util.cpp:



Namespaces

- prior_hessian
- prior_hessian::constants

Variables

- const double prior_hessian::constants::sqrt2 = std::sqrt(2.)
- const double prior_hessian::constants::sqrt2_inv = 1./std::sqrt(2.)
- const double prior_hessian::constants::sqrt2pi = std::sqrt(2.∗arma::datum::pi)
- const double prior_hessian::constants::sqrt2pi_inv = 1./std::sqrt(2.∗arma::datum::pi)
- const double prior_hessian::constants::log2pi = std::log(2.∗arma::datum::pi)

8.37.1 Detailed Description

Utilities.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

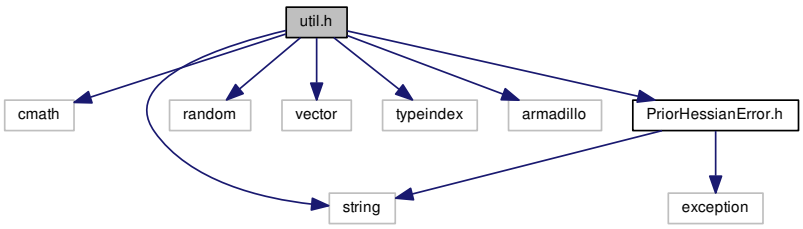
2017-2019

8.38 util.h File Reference

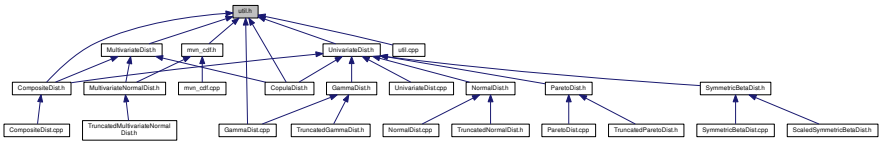
Utilities and namespace globals.

```
#include <cmath>
#include <string>
#include <random>
#include <vector>
#include <typeindex>
#include <armadillo>
#include "PriorHessianError.h"
```

Include dependency graph for util.h:



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



Namespaces

- prior_hessian
- prior_hessian::constants

Typedefs

- using prior_hessian::IdxT = arma::uword
- using prior_hessian::UVecT = arma::Col< IdxT >
- using prior_hessian::VecT = arma::Col< double >
- using prior_hessian::MatT = arma::Mat< double >
- using prior_hessian::StringVecT = std::vector< std::string >
- using prior_hessian::TypeInfoVecT = std::vector< std::type_index >

Functions

- template<class T >
  T prior_hessian::square (T t)

8.38.1 Detailed Description

Utilities and namespace globals.

Author

Mark J. Olah (mjo@cs.unm DOT edu)

Date

2017-2019

Index

_scaled_lbound
prior_hessian::ScaledDist, 343
_scaled_ubound
prior_hessian::ScaledDist, 343
_truncated
prior_hessian::TruncatedDist, 375
prior_hessian::TruncatedMultivariateDist, 381
_truncated_lbound
prior_hessian::TruncatedDist, 375
prior_hessian::TruncatedMultivariateDist, 381
_truncated_ubound
prior_hessian::TruncatedDist, 376
prior_hessian::TruncatedMultivariateDist, 382

AMHCopula
prior_hessian::AMHCopula, 49
AMHCopula.h, 395
ASSERT_SETUP
PriorHessianError.h, 424
adaptable_bounds
prior_hessian::detail::dist_adaptor_traits, 172
prior_hessian::detail::dist_adaptor_traits< Copula↩
DistImpl::CopulaDist< CopulaTemplate, Dist↩
Ts... > >, 173
prior_hessian::detail::dist_adaptor_traits< Gamma↩
Dist >, 174
prior_hessian::detail::dist_adaptor_traits< Multivariate↩
NormalDist< Ndim > >, 175
prior_hessian::detail::dist_adaptor_traits< Normal↩
Dist >, 176
prior_hessian::detail::dist_adaptor_traits< ParetoDist
>, 177
prior_hessian::detail::dist_adaptor_traits< Scaled↩
SymmetricBetaDist >, 178
prior_hessian::detail::dist_adaptor_traits< Symmetric↩
BetaDist >, 179
prior_hessian::detail::dist_adaptor_traits< Truncated↩
GammaDist >, 180
prior_hessian::detail::dist_adaptor_traits< Truncated↩
MultivariateNormalDist< Ndim > >, 181
prior_hessian::detail::dist_adaptor_traits< Truncated↩
NormalDist >, 182
prior_hessian::detail::dist_adaptor_traits< Truncated↩
ParetoDist >, 183
alpha
prior_hessian::ParetoDist, 287
AnyRngT
prior_hessian::CompositeDist, 93
append_global_lbound
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
MultivariateDist > >, 69
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
UnivariateDist > >, 80
append_global_ubound
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
MultivariateDist > >, 69
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
UnivariateDist > >, 80
append_lbound
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
MultivariateDist > >, 70
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
UnivariateDist > >, 81
append_param_names
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
MultivariateDist > >, 70
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
UnivariateDist > >, 81
append_params
prior_hessian::AMHCopula, 49
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
MultivariateDist > >, 71
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
UnivariateDist > >, 82
prior_hessian::MultivariateNormalDist, 224
append_params_iter
prior_hessian::NormalDist, 252
append_params_lbound
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
MultivariateDist > >, 71
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
UnivariateDist > >, 82
append_params_ubound
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
MultivariateDist > >, 71
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
UnivariateDist > >, 82
append_sample

prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 72
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 82
append_ubound
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 72
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 83
ArchimedeanCopula.h, 396

BaseDist.h, 397
beta
prior_hessian::SymmetricBetaDist, 347
bounded
prior_hessian, 16
bounds_adapted_dist
prior_hessian::detail::dist_adaptor_traits, 171
prior_hessian::detail::dist_adaptor_traits< Copula↩
DistImpl::CopulaDist< CopulaTemplate, Dist↩
Ts... > >, 172
prior_hessian::detail::dist_adaptor_traits< Gamma↩
Dist >, 173
prior_hessian::detail::dist_adaptor_traits< Multivariate↩
NormalDist< Ndim > >, 174
prior_hessian::detail::dist_adaptor_traits< Normal↩
Dist >, 175
prior_hessian::detail::dist_adaptor_traits< ParetoDist >, 176
prior_hessian::detail::dist_adaptor_traits< Scaled↩
SymmetricBetaDist >, 177
prior_hessian::detail::dist_adaptor_traits< Symmetric↩
BetaDist >, 178
prior_hessian::detail::dist_adaptor_traits< Truncated↩
GammaDist >, 179
prior_hessian::detail::dist_adaptor_traits< Truncated↩
MultivariateNormalDist< Ndim > >, 180
prior_hessian::detail::dist_adaptor_traits< Truncated↩
NormalDist >, 181
prior_hessian::detail::dist_adaptor_traits< Truncated↩
ParetoDist >, 182
bounds_pdf_integral
prior_hessian::TruncatedDist, 376
prior_hessian::TruncatedMultivariateDist, 382
BoundsAdaptedDist.h, 398
BoundsAdaptedDistT
prior_hessian, 14

call_in_order
prior_hessian::meta, 40
cdf
prior_hessian::AMHCopula, 49
prior_hessian::CompositeDist, 95
prior_hessian::CopulaDistImpl::CopulaDist, 141
prior_hessian::GammaDist, 190
prior_hessian::MultivariateNormalDist, 224, 225
prior_hessian::NormalDist, 252
prior_hessian::ParetoDist, 288
prior_hessian::ScaledDist, 331
prior_hessian::SymmetricBetaDist, 347
prior_hessian::TruncatedDist, 366
prior_hessian::TruncatedMultivariateDist, 379
prior_hessian::UpperTruncatedDist, 388
cdf_from_iter
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 72
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 83
check_bounds
prior_hessian::CopulaDistImpl::CopulaDist, 141
prior_hessian::GammaDist, 191
prior_hessian::MultivariateDist, 219
prior_hessian::MultivariateNormalDist, 225
prior_hessian::NormalDist, 252
prior_hessian::ParetoDist, 289
prior_hessian::SymmetricBetaDist, 348
prior_hessian::UnivariateDist, 384
check_copula_theta
prior_hessian::CopulaDistImpl::CopulaDist, 141
check_lbound
prior_hessian::ParetoDist, 290
check_mu
prior_hessian::MultivariateNormalDist, 225
check_params
prior_hessian::AMHCopula, 50
prior_hessian::CompositeDist, 95
prior_hessian::CopulaDistImpl::CopulaDist, 141
prior_hessian::GammaDist, 191
prior_hessian::MultivariateNormalDist, 225, 226
prior_hessian::NormalDist, 253
prior_hessian::ParetoDist, 290, 291
prior_hessian::SymmetricBetaDist, 348
check_params_iter
prior_hessian::AMHCopula, 50
prior_hessian::CopulaDistImpl::CopulaDist, 142
prior_hessian::GammaDist, 192
prior_hessian::MultivariateNormalDist, 227
prior_hessian::NormalDist, 254
prior_hessian::ParetoDist, 292
prior_hessian::SymmetricBetaDist, 349
check_sigma
prior_hessian::MultivariateNormalDist, 228
check_theta

prior_hessian::AMHCopula, 51
clear
prior_hessian::CompositeDist, 95
component_names
prior_hessian::CompositeDist, 96
component_types
prior_hessian::CompositeDist, 96
ComponentDistAdaptor
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 69
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 80
ComponentDistT
prior_hessian::CompositeDist, 93
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 69
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 80
CompositeDist
prior_hessian::CompositeDist, 93, 94
CompositeDist.cpp, 400
CompositeDist.h, 401
compute_quadratic_from_symmetric
prior_hessian::helpers, 33
compute_truncated_pdf_integral
prior_hessian::TruncatedMultivariateDist, 379
condition
prior_hessian::IndexError, 207
prior_hessian::InvalidOperationError, 210
prior_hessian::NotImplementedError, 274
prior_hessian::NumericalOverflowError, 276
prior_hessian::ParameterNameError, 278
prior_hessian::ParameterNameUniquenessError, 280
prior_hessian::ParameterSizeError, 282
prior_hessian::ParameterValueError, 284
prior_hessian::PriorHessianError, 321
prior_hessian::RuntimeConvergenceError, 323
prior_hessian::RuntimeSamplingError, 325
prior_hessian::RuntimeTypeError, 327
constexpr_accumulate
prior_hessian::meta, 40
ConstructableIf
prior_hessian::meta, 37
ConstructableIfAllAreNotTupleAndAreNotT
prior_hessian::meta, 37
ConstructableIfAllDistsAreBoundedT
prior_hessian::meta, 37
ConstructableIfInstantiatedFromNumericT
prior_hessian::meta, 37
ConstructableIfIsCopulaT
prior_hessian::meta, 37
ConstructableIfIsSuperClassForAllT
prior_hessian::meta, 37
ConstructableIfIsSuperClassT
prior_hessian::meta, 37
ConstructableIfIsTemplateForAllT
prior_hessian::meta, 37
ConstructableIfNot
prior_hessian::meta, 37
ConstructableIfNotAllDistsAreBoundedT
prior_hessian::meta, 38
ConstructableIfNotSelfT
prior_hessian::meta, 38
convert_from_unitary_coords
prior_hessian::ScaledDist, 331
convert_to_unitary_coords
prior_hessian::ScaledDist, 332
copula.h, 403
CopulaDist
prior_hessian, 14
prior_hessian::CopulaDistImpl::CopulaDist, 140
CopulaDist.h, 404
CopulaT
prior_hessian::CopulaDistImpl::CopulaDist, 139

d1_igen
prior_hessian::AMHCopula, 51
d1_igen_ui
prior_hessian::ArchimedeanCopula::D2_IGenTerms, 162
prior_hessian::ArchimedeanCopula::D_IGenTerms, 167
d2_igen_ui
prior_hessian::ArchimedeanCopula::D2_IGenTerms, 162
DEBUG_ASSERT
PriorHessianError.h, 424
ddim_gen
prior_hessian::AMHCopula, 52
DerivedFrom
prior_hessian::meta, 38
dim_variables
prior_hessian::CompositeDist, 97
DistTraitsT
prior_hessian::detail, 30
donnelly_bvn_cdf
prior_hessian, 16
donnelly_bvn_integral
prior_hessian, 16
donnelly_bvn_integral_orig
prior_hessian, 17

EnableIfAllAreNotTupleT
prior_hessian::meta, 38

EnableIfInstantiatedFromNumericT
prior_hessian::meta, 38
EnableIfInstantiatedFromT
prior_hessian::meta, 38
EnableIfIsNotTupleAndIsNotSelfT
prior_hessian::meta, 38
EnableIfIsNotTupleT
prior_hessian::meta, 38
EnableIfIsSuperclassOfAllT
prior_hessian::meta, 38
EnableIfIsTemplateForAllT
prior_hessian::meta, 39
EnableIfNonEmpty
prior_hessian::meta, 39
EnableIfNotInstantiatedFromT
prior_hessian::meta, 39
EnableIfNotIsSelfT
prior_hessian::meta, 39
EnableIfSubclassOfNumericTemplateT
prior_hessian::meta, 39
EnableIfSubclassT
prior_hessian::meta, 39
eta_0n_1n_t
prior_hessian::ArchimedeanCopula::D2Theta_↩
GenTerms, 164
prior_hessian::ArchimedeanCopula::DTheta_Gen↩
Terms, 183
eta_n_np1_t
prior_hessian::ArchimedeanCopula::D2_GenTerms,
161
prior_hessian::ArchimedeanCopula::D_GenTerms,
166
eulerian_polynomial
prior_hessian, 18
prior_hessian::detail, 30
prior_hessian::polylog, 43
eulerian_polynomial< 0 >
prior_hessian::polylog, 43
eulerian_polynomial< 1 >
prior_hessian::polylog, 43
eulerian_polynomial< 2 >
prior_hessian::polylog, 43
eulerian_polynomial< 3 >
prior_hessian::polylog, 43
eulerian_polynomial< 4 >
prior_hessian::polylog, 44
eulerian_polynomial< 5 >
prior_hessian::polylog, 44
eulerian_polynomial< 6 >
prior_hessian::polylog, 44
eulerian_polynomial< 7 >
prior_hessian::polylog, 44
eulerian_polynomial< 8 >
prior_hessian::polylog, 44
eulerian_polynomial< 9 >
prior_hessian::polylog, 44
EulerianPolynomial.h, 405

GCC_VERSION
Meta.h, 411
GammaDist
prior_hessian::GammaDist, 189, 190
GammaDist.cpp, 406
GammaDist.h, 407
gen
prior_hessian::AMHCopula, 53
get_copula_theta
prior_hessian::CopulaDistImpl::CopulaDist, 142
get_dim_variable_index
prior_hessian::CompositeDist, 97
get_dist_tuple
prior_hessian::CompositeDist, 98
get_param
prior_hessian::GammaDist, 192
prior_hessian::MultivariateNormalDist, 228
prior_hessian::NormalDist, 255
prior_hessian::ParetoDist, 293
prior_hessian::SymmetricBetaDist, 350
get_param_index
prior_hessian::CompositeDist, 98
get_param_value
prior_hessian::CompositeDist, 98
global_lbound
prior_hessian::CompositeDist, 99
prior_hessian::CopulaDistImpl::CopulaDist, 143
prior_hessian::ParetoDist, 294
prior_hessian::ScaledDist, 333
prior_hessian::TruncatedDist, 366
prior_hessian::TruncatedMultivariateDist, 380
global_ubound
prior_hessian::CompositeDist, 99
prior_hessian::CopulaDistImpl::CopulaDist, 143
prior_hessian::ScaledDist, 333
prior_hessian::TruncatedDist, 366
prior_hessian::TruncatedMultivariateDist, 380
prior_hessian::UpperTruncatedDist, 388
grad
prior_hessian::AMHCopula, 53
prior_hessian::CompositeDist, 100
prior_hessian::CopulaDistImpl::CopulaDist, 144
prior_hessian::GammaDist, 193
prior_hessian::MultivariateNormalDist, 228
prior_hessian::NormalDist, 255
prior_hessian::ParetoDist, 294
prior_hessian::SymmetricBetaDist, 350
grad2
prior_hessian::AMHCopula, 54
prior_hessian::CompositeDist, 101

prior_hessian::CopulaDistImpl::CopulaDist, 145
prior_hessian::GammaDist, 193
prior_hessian::MultivariateNormalDist, 229
prior_hessian::NormalDist, 256
prior_hessian::ParetoDist, 295
prior_hessian::SymmetricBetaDist, 351
grad2_accumulate
prior_hessian::CompositeDist, 102
grad2_accumulate_idx
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 73
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 84
grad_accumulate
prior_hessian::CompositeDist, 103
grad_accumulate_idx
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 73
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 84
grad_grad2_accumulate
prior_hessian::CompositeDist, 103
prior_hessian::CopulaDistImpl::CopulaDist, 145
prior_hessian::GammaDist, 193
prior_hessian::MultivariateNormalDist, 230
prior_hessian::NormalDist, 257
prior_hessian::ParetoDist, 296
prior_hessian::SymmetricBetaDist, 351
grad_grad2_accumulate_idx
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 74
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 84
grad_hess_accumulate
prior_hessian::CompositeDist, 104
prior_hessian::CopulaDistImpl::CopulaDist, 145
prior_hessian::MultivariateNormalDist, 231
grad_hess_accumulate_idx
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 74
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 85

has_dim_variable
prior_hessian::CompositeDist, 104
has_param
prior_hessian::CompositeDist, 105
hess
prior_hessian::AMHCopula, 54
prior_hessian::CompositeDist, 105
prior_hessian::CopulaDistImpl::CopulaDist, 146
prior_hessian::MultivariateNormalDist, 231
hess_accumulate
prior_hessian::CompositeDist, 106
hess_accumulate_idx
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 74
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 85

icdf
prior_hessian::GammaDist, 194
prior_hessian::NormalDist, 257
prior_hessian::ParetoDist, 297
prior_hessian::ScaledDist, 333
prior_hessian::SymmetricBetaDist, 352
prior_hessian::TruncatedDist, 367
prior_hessian::UpperTruncatedDist, 388
IdxT
prior_hessian, 14
ieta_21_ui
prior_hessian::ArchimedeanCopula::D2_IGenTerms, 162
prior_hessian::ArchimedeanCopula::D_IGenTerms, 167
igen
prior_hessian::AMHCopula, 55
igen_sum
prior_hessian::AMHCopula, 55
in_bounds
prior_hessian::CompositeDist, 106
prior_hessian::GammaDist, 194
prior_hessian::MultivariateNormalDist, 232
prior_hessian::NormalDist, 258
prior_hessian::ParetoDist, 298
prior_hessian::SymmetricBetaDist, 352
prior_hessian::TruncatedMultivariateDist, 380
in_bounds_all
prior_hessian::CompositeDist, 107
IndexError
prior_hessian::IndexError, 207
initialize
prior_hessian::CompositeDist, 107–109
initialize_copula
prior_hessian::CopulaDistImpl::CopulaDist, 146
initialize_marginals
prior_hessian::CopulaDistImpl::CopulaDist, 147
InvalidOperationError

prior_hessian::InvalidOperationError, 209
is_empty
prior_hessian::CompositeDist, 110
ixi_1_ui
prior_hessian::ArchimedeanCopula::D2_IGenTerms, 162

lbound
prior_hessian::CompositeDist, 110
prior_hessian::CopulaDistImpl::CopulaDist, 147
prior_hessian::GammaDist, 195
prior_hessian::MultivariateNormalDist, 233
prior_hessian::NormalDist, 258
prior_hessian::ParetoDist, 299
prior_hessian::ScaledDist, 334
prior_hessian::SymmetricBetaDist, 353
prior_hessian::TruncatedDist, 367
prior_hessian::TruncatedMultivariateDist, 380
lbound_cdf
prior_hessian::TruncatedDist, 376
prior_hessian::TruncatedMultivariateDist, 382
llh
prior_hessian::AMHCopula, 56
prior_hessian::CompositeDist, 111
prior_hessian::CopulaDistImpl::CopulaDist, 147
prior_hessian::GammaDist, 195
prior_hessian::MultivariateNormalDist, 234
prior_hessian::NormalDist, 259
prior_hessian::ParetoDist, 301
prior_hessian::ScaledDist, 335
prior_hessian::SymmetricBetaDist, 353
prior_hessian::TruncatedDist, 368
prior_hessian::TruncatedMultivariateDist, 380
prior_hessian::UpperTruncatedDist, 389
llh_components
prior_hessian::CompositeDist, 111
llh_from_iter
prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 75
prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 86
llh_scaling_const
prior_hessian::ScaledDist, 343
llh_truncation_const
prior_hessian::TruncatedDist, 376
prior_hessian::TruncatedMultivariateDist, 382
log2pi
prior_hessian::constants, 29
log_dn_gen_t
prior_hessian::ArchimedeanCopula::D2_GenTerms, 161
prior_hessian::ArchimedeanCopula::D2Theta_GenTerms, 164
prior_hessian::ArchimedeanCopula::D_GenTerms, 166
prior_hessian::ArchimedeanCopula::DTheta_GenTerms, 183
logical_and_in_order
prior_hessian::meta, 41

MCMCData
prior_hessian::mcmc::MCMCData, 217
make_adapted_bounded_dist
prior_hessian, 18, 19
make_adapted_bounded_dist_tuple
prior_hessian, 19, 20
prior_hessian::detail, 30, 31
make_bounded_gamma_dist
prior_hessian, 20
make_bounded_multivariate_normal_dist
prior_hessian, 20
make_bounded_normal_dist
prior_hessian, 20
make_bounded_pareto_dist
prior_hessian, 20
make_component_dist
prior_hessian::CompositeDist, 112
make_component_dist_tuple
prior_hessian::CompositeDist, 113, 114
make_copula_dist
prior_hessian, 21
make_scaled_symmetric_beta_dist
prior_hessian, 21
make_zero_grad
prior_hessian::CompositeDist, 115
make_zero_hess
prior_hessian::CompositeDist, 115
MarginalDistTupleT
prior_hessian::CopulaDistImpl::CopulaDist, 139
MarginalDistT
prior_hessian::CopulaDistImpl::CopulaDist, 139
MatT
prior_hessian, 14
mc_mvn_cdf
prior_hessian, 21
mc_mvn_cdf_core
prior_hessian, 21
mc_mvn_integral
prior_hessian, 22
mean
prior_hessian::GammaDist, 196
prior_hessian::MultivariateNormalDist, 234
prior_hessian::NormalDist, 259
prior_hessian::ParetoDist, 302
prior_hessian::ScaledDist, 335

prior_hessian::SymmetricBetaDist, 354
prior_hessian::TruncatedDist, 368
prior_hessian::TruncatedMultivariateDist, 380
prior_hessian::UpperTruncatedDist, 389
median
prior_hessian::GammaDist, 196
prior_hessian::NormalDist, 260
prior_hessian::ParetoDist, 304
prior_hessian::ScaledDist, 336
prior_hessian::SymmetricBetaDist, 354
prior_hessian::TruncatedDist, 368
prior_hessian::UpperTruncatedDist, 389
Meta.h, 408
GCC_VERSION, 411
PRIOR_HESSIAN_META_CONSTEXPR, 411
PRIOR_HESSIAN_META_HAS_CONSTEXPR, 411
min
prior_hessian::ParetoDist, 305
min_bounds_pdf_integral
prior_hessian::TruncatedDist, 376
prior_hessian::TruncatedMultivariateDist, 382
mode
prior_hessian::MultivariateNormalDist, 235
mu
prior_hessian::MultivariateNormalDist, 235
prior_hessian::NormalDist, 260
MultivariateDist
prior_hessian::MultivariateDist, 219
MultivariateDist.h, 412
MultivariateNormalDist
prior_hessian::MultivariateNormalDist, 223
MultivariateNormalDist.h, 413
mutex
prior_hessian::mcmc::MCMCData, 218
mvn_cdf.cpp, 414
mvn_cdf.h, 415
mvn_cdf_genz
prior_hessian::genz, 32
mvndst_
prior_hessian::genz::fortran, 33

NdimMatT
prior_hessian::AMHCopula, 48
prior_hessian::CopulaDistImpl::CopulaDist, 139
prior_hessian::MultivariateNormalDist, 222
NdimVecT
prior_hessian::AMHCopula, 48
prior_hessian::CopulaDistImpl::CopulaDist, 140
prior_hessian::MultivariateNormalDist, 222
prior_hessian::mcmc::MCMCData, 217
NormalDist
prior_hessian::NormalDist, 251
NormalDist.cpp, 417
NormalDist.h, 418
NotImplementedError
prior_hessian::NotImplementedError, 273
NparamsVecT
prior_hessian::CopulaDistImpl::CopulaDist, 140
prior_hessian::GammaDist, 189
prior_hessian::MultivariateNormalDist, 222
prior_hessian::NormalDist, 250
prior_hessian::ParetoDist, 286
prior_hessian::SymmetricBetaDist, 346
nsample
prior_hessian::mcmc::MCMCData, 218
num_components
prior_hessian::CompositeDist, 115
prior_hessian::CopulaDistImpl::CopulaDist, 148
num_dim
prior_hessian::AMHCopula, 57
prior_hessian::CompositeDist, 116
prior_hessian::CopulaDistImpl::CopulaDist, 148
prior_hessian::GammaDist, 197
prior_hessian::MultivariateNormalDist, 236
prior_hessian::NormalDist, 261
prior_hessian::ParetoDist, 306
prior_hessian::SymmetricBetaDist, 354
prior_hessian::UnivariateDist, 384
num_dim_components
prior_hessian::CompositeDist, 117
num_params
prior_hessian::AMHCopula, 57
prior_hessian::CompositeDist, 117
prior_hessian::CopulaDistImpl::CopulaDist, 149
prior_hessian::GammaDist, 197
prior_hessian::MultivariateNormalDist, 237
prior_hessian::NormalDist, 261
prior_hessian::ParetoDist, 306
prior_hessian::SymmetricBetaDist, 354
prior_hessian::TruncatedDist, 369
num_params_components
prior_hessian::CompositeDist, 117
NumericalOverflowError
prior_hessian::NumericalOverflowError, 275

operator bool
prior_hessian::CompositeDist, 118
operator!=
prior_hessian::AMHCopula, 57
prior_hessian::CompositeDist, 118
prior_hessian::CopulaDistImpl::CopulaDist, 149
prior_hessian::GammaDist, 197
prior_hessian::MultivariateNormalDist, 237
prior_hessian::NormalDist, 261
prior_hessian::ParetoDist, 306
prior_hessian::ScaledDist, 336
prior_hessian::SymmetricBetaDist, 355
prior_hessian::TruncatedDist, 369

prior_hessian::TruncatedMultivariateDist, 380
prior_hessian::UpperTruncatedDist, 390
operator<<
prior_hessian, 23
operator=
prior_hessian::CompositeDist, 118, 119
prior_hessian::mcmc::MCMCData, 217
operator==
prior_hessian::AMHCopula, 57
prior_hessian::CompositeDist, 119
prior_hessian::CopulaDistImpl::CopulaDist, 151
prior_hessian::GammaDist, 197
prior_hessian::MultivariateNormalDist, 238
prior_hessian::NormalDist, 262
prior_hessian::ParetoDist, 307
prior_hessian::ScaledDist, 337
prior_hessian::SymmetricBetaDist, 355
prior_hessian::TruncatedDist, 369
prior_hessian::TruncatedMultivariateDist, 380
prior_hessian::UpperTruncatedDist, 390
owen_b_integral
prior_hessian, 24
owen_bvn_cdf
prior_hessian, 25
owen_t_integral
prior_hessian, 26

PRIOR_HESSIAN_META_CONSTEXPR
Meta.h, 411
PRIOR_HESSIAN_META_HAS_CONSTEXPR
Meta.h, 411
param_lbound
prior_hessian::AMHCopula, 58
prior_hessian::CopulaDistImpl::CopulaDist, 152
prior_hessian::GammaDist, 198
prior_hessian::MultivariateNormalDist, 238
prior_hessian::NormalDist, 262
prior_hessian::ParetoDist, 307
prior_hessian::SymmetricBetaDist, 355
param_names
prior_hessian::AMHCopula, 58
prior_hessian::CompositeDist, 119
prior_hessian::CopulaDistImpl::CopulaDist, 152
prior_hessian::GammaDist, 198
prior_hessian::MultivariateNormalDist, 239
prior_hessian::NormalDist, 263
prior_hessian::ParetoDist, 308
prior_hessian::SymmetricBetaDist, 356
param_ubound
prior_hessian::AMHCopula, 58
prior_hessian::CopulaDistImpl::CopulaDist, 153
prior_hessian::GammaDist, 198
prior_hessian::MultivariateNormalDist, 239
prior_hessian::NormalDist, 263
prior_hessian::ParetoDist, 308
prior_hessian::SymmetricBetaDist, 356
ParameterNameError
prior_hessian::ParameterNameError, 277
ParameterNameUniquenessError
prior_hessian::ParameterNameUniquenessError, 279
ParameterSizeError
prior_hessian::ParameterSizeError, 281
ParameterValueError
prior_hessian::ParameterValueError, 283
params
prior_hessian::CompositeDist, 120
prior_hessian::CopulaDistImpl::CopulaDist, 153
prior_hessian::GammaDist, 198
prior_hessian::MultivariateNormalDist, 239
prior_hessian::NormalDist, 263
prior_hessian::ParetoDist, 308
prior_hessian::SymmetricBetaDist, 356
params_components
prior_hessian::CompositeDist, 121
params_lbound
prior_hessian::CompositeDist, 122
params_ubound
prior_hessian::CompositeDist, 122
ParetoDist
prior_hessian::ParetoDist, 287
ParetoDist.cpp, 419
ParetoDist.h, 420
pdf
prior_hessian::AMHCopula, 58
prior_hessian::CompositeDist, 123
prior_hessian::CopulaDistImpl::CopulaDist, 154
prior_hessian::GammaDist, 199
prior_hessian::MultivariateNormalDist, 240
prior_hessian::NormalDist, 264
prior_hessian::ParetoDist, 309
prior_hessian::ScaledDist, 337
prior_hessian::SymmetricBetaDist, 357
prior_hessian::TruncatedDist, 370
prior_hessian::TruncatedMultivariateDist, 380
prior_hessian::UpperTruncatedDist, 391
pdf_from_iter
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 75
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 86
PolyLog.h, 421
polylog
prior_hessian::polylog, 44
polylog< 1 >
prior_hessian::polylog, 45

prior_hessian, 12
bounded, 16
BoundsAdaptedDistT, 14
CopulaDist, 14
donnelly_bvn_cdf, 16
donnelly_bvn_integral, 16
donnelly_bvn_integral_orig, 17
eulerian_polynomial, 18
IdxT, 14
make_adapted_bounded_dist, 18, 19
make_adapted_bounded_dist_tuple, 19, 20
make_bounded_gamma_dist, 20
make_bounded_multivariate_normal_dist, 20
make_bounded_normal_dist, 20
make_bounded_pareto_dist, 20
make_copula_dist, 21
make_scaled_symmetric_beta_dist, 21
MatT, 14
mc_mvn_cdf, 21
mc_mvn_cdf_core, 21
mc_mvn_integral, 22
operator<<, 23
owen_b_integral, 24
owen_bvn_cdf, 25
owen_t_integral, 26
ScaledSymmetricBetaDist, 14
square, 27
StringVecT, 15
TruncatedGammaDist, 15
TruncatedMultivariateNormalDist, 15
TruncatedNormalDist, 15
TruncatedParetoDist, 15
TypeInfoVecT, 15
UVecT, 15
unit_normal_cdf, 27
unit_normal_icdf, 28
VecT, 15
prior_hessian::AMHCopula
AMHCopula, 49
append_params, 49
cdf, 49
check_params, 50
check_params_iter, 50
check_theta, 51
d1_igen, 51
ddim_gen, 52
gen, 53
grad, 53
grad2, 54
hess, 54
igen, 55
igen_sum, 55
llh, 56
NdimMatT, 48
NdimVecT, 48
num_dim, 57
num_params, 57
operator!=, 57
operator==, 57
param_lbound, 58
param_names, 58
param_ubound, 58
pdf, 58
rllh, 59
rllh_const, 59
rllh_d2theta_accumulate, 60
rllh_dtheta_accumulate, 60
rllh_grad_accumulate, 60
rllh_grad_grad2_accumulate, 61
rllh_grad_hess_accumulate, 61
sample, 61
set_params, 62
set_params_iter, 62
set_theta, 63
theta, 64
prior_hessian::AMHCopula< Ndim >, 46
prior_hessian::ArchimedeanCopula, 65
prior_hessian::ArchimedeanCopula::D2_GenTerms, 159
eta_n_np1_t, 161
log_dn_gen_t, 161
xi_n_t, 161
prior_hessian::ArchimedeanCopula::D2_IGenTerms, 161
d1_igen_ui, 162
d2_igen_ui, 162
ieta_21_ui, 162
ixi_1_ui, 162
prior_hessian::ArchimedeanCopula::D2Theta_GenTerms, 163
eta_0n_1n_t, 164
log_dn_gen_t, 164
xi_0n_t, 164
prior_hessian::ArchimedeanCopula::D2Theta_IGen↩ Terms, 164
sum_d10_igen_u, 165
sum_d20_igen_u, 165
sum_ieta_01_11_u, 165
sum_ixi_01_u, 165
sum_log_d1_igen_u, 165
prior_hessian::ArchimedeanCopula::D_GenTerms, 166
eta_n_np1_t, 166
log_dn_gen_t, 166
prior_hessian::ArchimedeanCopula::D_IGenTerms, 167
d1_igen_ui, 167
ieta_21_ui, 167
prior_hessian::ArchimedeanCopula::DTheta_GenTerms, 183
eta_0n_1n_t, 183
log_dn_gen_t, 183

prior_hessian::ArchimedeanCopula::DTheta_IGenTerms, 184
sum_d10_igen_u, 184
sum_ieta_01_11_u, 184
sum_log_d1_igen_u, 185
prior_hessian::BaseDist, 66
prior_hessian::CompositeDist, 89
AnyRngT, 93
cdf, 95
check_params, 95
clear, 95
component_names, 96
component_types, 96
ComponentDistT, 93
CompositeDist, 93, 94
dim_variables, 97
get_dim_variable_index, 97
get_dist_tuple, 98
get_param_index, 98
get_param_value, 98
global_lbound, 99
global_ubound, 99
grad, 100
grad2, 101
grad2_accumulate, 102
grad_accumulate, 103
grad_grad2_accumulate, 103
grad_hess_accumulate, 104
has_dim_variable, 104
has_param, 105
hess, 105
hess_accumulate, 106
in_bounds, 106
in_bounds_all, 107
initialize, 107–109
is_empty, 110
lbound, 110
llh, 111
llh_components, 111
make_component_dist, 112
make_component_dist_tuple, 113, 114
make_zero_grad, 115
make_zero_hess, 115
num_components, 115
num_dim, 116
num_dim_components, 117
num_params, 117
num_params_components, 117
operator bool, 118
operator!=, 118
operator=, 118, 119
operator==, 119
param_names, 119
params, 120
params_components, 121
params_lbound, 122
params_ubound, 122
pdf, 123
rename_dim_variable, 123
rename_param, 124
rllh, 124
rllh_components, 125
sample, 126, 127
set_bounds, 127
set_component_names, 128
set_dim_variables, 129
set_lbound, 129
set_param_names, 130
set_param_value, 131
set_params, 132
set_ubound, 132
ubound, 133
prior_hessian::CompositeDist::ComponentDistAdaptor, 92, 403
prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 67
append_global_lbound, 69
append_global_ubound, 69
append_lbound, 70
append_param_names, 70
append_params, 71
append_params_lbound, 71
append_params_ubound, 71
append_sample, 72
append_ubound, 72
cdf_from_iter, 72
ComponentDistAdaptor, 69
ComponentDistT, 69
grad2_accumulate_idx, 73
grad_accumulate_idx, 73
grad_grad2_accumulate_idx, 74
grad_hess_accumulate_idx, 74
hess_accumulate_idx, 74
llh_from_iter, 75
pdf_from_iter, 75
rllh_from_iter, 76
set_bounds_from_iter, 76
set_lbound_from_iter, 77
set_ubound_from_iter, 77
prior_hessian::CompositeDist::ComponentDistAdaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 78
append_global_lbound, 80
append_global_ubound, 80
append_lbound, 81
append_param_names, 81
append_params, 82

append_params_lbound, 82
append_params_ubound, 82
append_sample, 82
append_ubound, 83
cdf_from_iter, 83
ComponentDistAdaptor, 80
ComponentDistT, 80
grad2_accumulate_idx, 84
grad_accumulate_idx, 84
grad_grad2_accumulate_idx, 84
grad_hess_accumulate_idx, 85
hess_accumulate_idx, 85
llh_from_iter, 86
pdf_from_iter, 86
rllh_from_iter, 87
set_bounds_from_iter, 87
set_lbound_from_iter, 88
set_ubound_from_iter, 88
prior_hessian::CopulaDistImpl, 29
prior_hessian::CopulaDistImpl::CopulaDist
cdf, 141
check_bounds, 141
check_copula_theta, 141
check_params, 141
check_params_iter, 142
CopulaDist, 140
CopulaT, 139
get_copula_theta, 142
global_lbound, 143
global_ubound, 143
grad, 144
grad2, 145
grad_grad2_accumulate, 145
grad_hess_accumulate, 145
hess, 146
initialize_copula, 146
initialize_marginals, 147
lbound, 147
llh, 147
MarginalDistTupleT, 139
MarginalDistT, 139
NdimMatT, 139
NdimVecT, 140
NparamsVecT, 140
num_components, 148
num_dim, 148
num_params, 149
operator!=, 149
operator==, 151
param_lbound, 152
param_names, 152
param_ubound, 153
params, 153
pdf, 154
rllh, 154
sample, 154
set_bounds, 155
set_copula_theta, 155
set_lbound, 156
set_params, 157
set_params_iter, 157
set_ubound, 157
ubound, 158
prior_hessian::CopulaDistImpl::CopulaDist< CopulaTemplate, MarginalDistTs >, 137
prior_hessian::GammaDist, 187
cdf, 190
check_bounds, 191
check_params, 191
check_params_iter, 192
GammaDist, 189, 190
get_param, 192
grad, 193
grad2, 193
grad_grad2_accumulate, 193
icdf, 194
in_bounds, 194
lbound, 195
llh, 195
mean, 196
median, 196
NparamsVecT, 189
num_dim, 197
num_params, 197
operator!=, 197
operator==, 197
param_lbound, 198
param_names, 198
param_ubound, 198
params, 198
pdf, 199
rllh, 199
sample, 199
scale, 200
set_param, 200
set_params, 201, 202
set_params_iter, 202
set_scale, 202
set_shape, 203
shape, 203
ubound, 204
prior_hessian::IndexError, 205
condition, 207
IndexError, 207
what, 207
what_, 207
what_str, 207
prior_hessian::InvalidOperationError, 208

condition, 210
InvalidOperationError, 209
what, 210
what_, 210
what_str, 210
prior_hessian::MultivariateDist, 218
check_bounds, 219
MultivariateDist, 219
prior_hessian::MultivariateNormalDist
append_params, 224
cdf, 224, 225
check_bounds, 225
check_mu, 225
check_params, 225, 226
check_params_iter, 227
check_sigma, 228
get_param, 228
grad, 228
grad2, 229
grad_grad2_accumulate, 230
grad_hess_accumulate, 231
hess, 231
in_bounds, 232
lbound, 233
llh, 234
mean, 234
mode, 235
mu, 235
MultivariateNormalDist, 223
NdimMatT, 222
NdimVecT, 222
NparamsVecT, 222
num_dim, 236
num_params, 237
operator!=, 237
operator==, 238
param_lbound, 238
param_names, 239
param_ubound, 239
params, 239
pdf, 240
rllh, 241
sample, 242
set_mu, 243
set_params, 243, 244
set_params_iter, 245
set_sigma, 245
sigma, 246
sigma_inv, 246
ubound, 247
prior_hessian::MultivariateNormalDist< Ndim >, 220
prior_hessian::NormalDist, 248
append_params_iter, 252
cdf, 252
check_bounds, 252
check_params, 253
check_params_iter, 254
get_param, 255
grad, 255
grad2, 256
grad_grad2_accumulate, 257
icdf, 257
in_bounds, 258
lbound, 258
llh, 259
mean, 259
median, 260
mu, 260
NormalDist, 251
NparamsVecT, 250
num_dim, 261
num_params, 261
operator!=, 261
operator==, 262
param_lbound, 262
param_names, 263
param_ubound, 263
params, 263
pdf, 264
rllh, 264
sample, 265
set_mu, 266
set_param, 266
set_params, 267, 268
set_params_iter, 269
set_sigma, 269
sigma, 270
ubound, 271
prior_hessian::NotImplementedError, 272
condition, 274
NotImplementedError, 273
what, 274
what_, 274
what_str, 274
prior_hessian::NumericalOverflowError, 274
condition, 276
NumericalOverflowError, 275
what, 276
what_, 276
what_str, 276
prior_hessian::ParameterNameError, 276
condition, 278
ParameterNameError, 277
what, 278
what_, 278
what_str, 278
prior_hessian::ParameterNameUniquenessError, 278
condition, 280

ParameterNameUniquenessError, 279
what, 280
what_, 280
what_str, 280
prior_hessian::ParameterSizeError, 280
condition, 282
ParameterSizeError, 281
what, 282
what_, 282
what_str, 282
prior_hessian::ParameterValueError, 282
condition, 284
ParameterValueError, 283
what, 284
what_, 284
what_str, 284
prior_hessian::ParetoDist, 284
alpha, 287
cdf, 288
check_bounds, 289
check_lbound, 290
check_params, 290, 291
check_params_iter, 292
get_param, 293
global_lbound, 294
grad, 294
grad2, 295
grad_grad2_accumulate, 296
icdf, 297
in_bounds, 298
lbound, 299
llh, 301
mean, 302
median, 304
min, 305
NparamsVecT, 286
num_dim, 306
num_params, 306
operator!=, 306
operator==, 307
param_lbound, 307
param_names, 308
param_ubound, 308
params, 308
ParetoDist, 287
pdf, 309
rllh, 311
sample, 312
set_alpha, 313
set_lbound, 314
set_min, 314
set_param, 315
set_params, 316
set_params_iter, 317
ubound, 318
prior_hessian::PriorHessianError, 319
condition, 321
PriorHessianError, 320
what, 321
what_, 321
what_str, 321
prior_hessian::RuntimeConvergenceError, 321
condition, 323
RuntimeConvergenceError, 322
what, 323
what_, 323
what_str, 323
prior_hessian::RuntimeSamplingError, 323
condition, 325
RuntimeSamplingError, 324
what, 325
what_, 325
what_str, 325
prior_hessian::RuntimeTypeError, 325
condition, 327
RuntimeTypeError, 326
what, 327
what_, 327
what_str, 327
prior_hessian::ScaledDist
_scaled_lbound, 343
_scaled_ubound, 343
cdf, 331
convert_from_unitary_coords, 331
convert_to_unitary_coords, 332
global_lbound, 333
global_ubound, 333
icdf, 333
lbound, 334
llh, 335
llh_scaling_const, 343
mean, 335
median, 336
operator!=, 336
operator==, 337
pdf, 337
sample, 338
ScaledDist, 329, 330
scaling_ratio, 343
set_bounds, 339
set_lbound, 340
set_ubound, 340
ubound, 341
unscaled_lbound, 342
unscaled_ubound, 342
prior_hessian::ScaledDist< Dist >, 327
prior_hessian::SymmetricBetaDist, 344
beta, 347

cdf, 347
check_bounds, 348
check_params, 348
check_params_iter, 349
get_param, 350
grad, 350
grad2, 351
grad_grad2_accumulate, 351
icdf, 352
in_bounds, 352
lbound, 353
llh, 353
mean, 354
median, 354
NparamsVecT, 346
num_dim, 354
num_params, 354
operator!=, 355
operator==, 355
param_lbound, 355
param_names, 356
param_ubound, 356
params, 356
pdf, 357
rllh, 357
sample, 357
set_beta, 358
set_param, 358
set_params, 359, 360
set_params_iter, 360
SymmetricBetaDist, 346
ubound, 361
prior_hessian::TruncatedDist
_truncated, 375
_truncated_lbound, 375
_truncated_ubound, 376
bounds_pdf_integral, 376
cdf, 366
global_lbound, 366
global_ubound, 366
icdf, 367
lbound, 367
lbound_cdf, 376
llh, 368
llh_truncation_const, 376
mean, 368
median, 368
min_bounds_pdf_integral, 376
num_params, 369
operator!=, 369
operator==, 369
pdf, 370
sample, 370
set_bounds, 371
set_lbound, 372
set_ubound, 373
truncated, 374
TruncatedDist, 363–365
ubound, 375
prior_hessian::TruncatedDist< Dist >, 361
prior_hessian::TruncatedMultivariateDist
_truncated, 381
_truncated_lbound, 381
_truncated_ubound, 382
bounds_pdf_integral, 382
cdf, 379
compute_truncated_pdf_integral, 379
global_lbound, 380
global_ubound, 380
in_bounds, 380
lbound, 380
lbound_cdf, 382
llh, 380
llh_truncation_const, 382
mean, 380
min_bounds_pdf_integral, 382
operator!=, 380
operator==, 380
pdf, 380
sample, 381
set_bounds, 381
set_lbound, 381
set_ubound, 381
truncated, 381
TruncatedMultivariateDist, 379
ubound, 381
prior_hessian::TruncatedMultivariateDist< Dist >, 377
prior_hessian::UnivariateDist, 383
check_bounds, 384
num_dim, 384
UnivariateDist, 384
prior_hessian::UpperTruncatedDist
cdf, 388
global_ubound, 388
icdf, 388
llh, 389
mean, 389
median, 389
operator!=, 390
operator==, 390
pdf, 391
sample, 391
set_bounds, 392
set_lbound, 392
set_ubound, 393
truncated, 394
ubound, 394
UpperTruncatedDist, 386, 387

prior_hessian::UpperTruncatedDist< Dist >, 385
prior_hessian::constants, 29
log2pi, 29
sqrt2, 29
sqrt2_inv, 29
sqrt2pi, 29
sqrt2pi_inv, 29
prior_hessian::detail, 30
DistTraitsT, 30
eulerian_polynomial, 30
make_adapted_bounded_dist_tuple, 30, 31
prior_hessian::detail::dist_adaptor_traits
adaptable_bounds, 172
bounds_adapted_dist, 171
prior_hessian::detail::dist_adaptor_traits< CopulaDist↩
Impl::CopulaDist< CopulaTemplate, DistTs... > >, 172
adaptable_bounds, 173
bounds_adapted_dist, 172
prior_hessian::detail::dist_adaptor_traits< Dist >, 171
prior_hessian::detail::dist_adaptor_traits< GammaDist >, 173
adaptable_bounds, 174
bounds_adapted_dist, 173
prior_hessian::detail::dist_adaptor_traits< MultivariateNormalDist< Ndim > >, 174
adaptable_bounds, 175
bounds_adapted_dist, 174
prior_hessian::detail::dist_adaptor_traits< NormalDist >, 175
adaptable_bounds, 176
bounds_adapted_dist, 175
prior_hessian::detail::dist_adaptor_traits< ParetoDist >, 176
adaptable_bounds, 177
bounds_adapted_dist, 176
prior_hessian::detail::dist_adaptor_traits< ScaledSymmetricBetaDist >, 177
adaptable_bounds, 178
bounds_adapted_dist, 177
prior_hessian::detail::dist_adaptor_traits< SymmetricBetaDist >, 178
adaptable_bounds, 179
bounds_adapted_dist, 178
prior_hessian::detail::dist_adaptor_traits< TruncatedGammaDist >, 179
adaptable_bounds, 180
bounds_adapted_dist, 179
prior_hessian::detail::dist_adaptor_traits< TruncatedMultivariateNormalDist< Ndim > >, 180
adaptable_bounds, 181
bounds_adapted_dist, 180
prior_hessian::detail::dist_adaptor_traits< TruncatedNormalDist >, 181
adaptable_bounds, 182
bounds_adapted_dist, 181
prior_hessian::detail::dist_adaptor_traits< TruncatedParetoDist >, 182
adaptable_bounds, 183
bounds_adapted_dist, 182
prior_hessian::eulerian_number< 0, M >, 186
prior_hessian::eulerian_number< N, M >, 185
prior_hessian::genz, 32
mvn_cdf_genz, 32
prior_hessian::genz::fortran, 33
mvndst_, 33
prior_hessian::helpers, 33
compute_quadratic_from_symmetric, 33
prior_hessian::mcmc, 34
prior_hessian::mcmc::MCMCData
MCMCData, 217
mutex, 218
NdimVecT, 217
nsample, 218
operator=, 217
rllh, 218
sample, 218
prior_hessian::mcmc::MCMCData< Ndim >, 216
prior_hessian::meta, 34
call_in_order, 40
constexpr_accumulate, 40
ConstructableIf, 37
ConstructableIfAllAreNotTupleAndAreNotT, 37
ConstructableIfAllDistsAreBoundedT, 37
ConstructableIfInstantiatedFromNumericT, 37
ConstructableIfIsCopulaT, 37
ConstructableIfIsSuperClassForAllT, 37
ConstructableIfIsSuperClassT, 37
ConstructableIfIsTemplateForAllT, 37
ConstructableIfNot, 37
ConstructableIfNotAllDistsAreBoundedT, 38
ConstructableIfNotSelfT, 38
DerivedFrom, 38
EnableIfAllAreNotTupleT, 38
EnableIfInstantiatedFromNumericT, 38
EnableIfInstantiatedFromT, 38
EnableIfIsNotTupleAndIsNotSelfT, 38
EnableIfIsNotTupleT, 38
EnableIfIsSuperclassOfAllT, 38
EnableIfIsTemplateForAllT, 39
EnableIfNonEmpty, 39
EnableIfNotInstantiatedFromT, 39
EnableIfNotIsSelfT, 39
EnableIfSubclassOfNumericTemplateT, 39
EnableIfSubclassT, 39
logical_and_in_order, 41
prod_in_order, 41
ReturnIfInstantiatedFromNumericT, 39

ReturnIfInstantiatedFromT, 39
ReturnIfNotInstantiatedFromT, 39
ReturnIfSubclassOfNumericTemplateT, 40
ReturnIfSubclassT, 40
ReturnIfT, 40
sum_in_order, 42
prior_hessian::meta::all_dists_are_bounded
value, 46
prior_hessian::meta::all_dists_are_bounded< DistTs >, 45
prior_hessian::meta::conjunction< B1 >, 135
prior_hessian::meta::conjunction< B1, Bn... >, 136
prior_hessian::meta::conjunction<... >, 134
prior_hessian::meta::disjunction< B1 >, 169
prior_hessian::meta::disjunction< B1, Bn... >, 170
prior_hessian::meta::disjunction<... >, 168
prior_hessian::meta::is_copula
value, 211
prior_hessian::meta::is_copula< T, U >, 210
prior_hessian::meta::is_numeric_template_of< class, typename >, 211
prior_hessian::meta::is_numeric_template_of< Class↩ NumericTemplate, ClassNumericTemplate< Is... > >, 212
prior_hessian::meta::is_subclass_of_numeric_template
value, 214
prior_hessian::meta::is_subclass_of_numeric_template< T, U >, 213
prior_hessian::meta::is_template_of< class, typename >, 214
prior_hessian::meta::is_template_of< ClassTemplate, ClassTemplate< Ts... > >, 215
prior_hessian::polylog, 43
eulerian_polynomial, 43
eulerian_polynomial< 0 >, 43
eulerian_polynomial< 1 >, 43
eulerian_polynomial< 2 >, 43
eulerian_polynomial< 3 >, 43
eulerian_polynomial< 4 >, 44
eulerian_polynomial< 5 >, 44
eulerian_polynomial< 6 >, 44
eulerian_polynomial< 7 >, 44
eulerian_polynomial< 8 >, 44
eulerian_polynomial< 9 >, 44
polylog, 44
polylog< 1 >, 45
PriorHessianError
prior_hessian::PriorHessianError, 320
PriorHessianError.h, 422
ASSERT_SETUP, 424
DEBUG_ASSERT, 424
prod_in_order
prior_hessian::meta, 41
README.md, 424
rename_dim_variable
prior_hessian::CompositeDist, 123
rename_param
prior_hessian::CompositeDist, 124
ReturnIfInstantiatedFromNumericT
prior_hessian::meta, 39
ReturnIfInstantiatedFromT
prior_hessian::meta, 39
ReturnIfNotInstantiatedFromT
prior_hessian::meta, 39
ReturnIfSubclassOfNumericTemplateT
prior_hessian::meta, 40
ReturnIfSubclassT
prior_hessian::meta, 40
ReturnIfT
prior_hessian::meta, 40
rllh
prior_hessian::AMHCopula, 59
prior_hessian::CompositeDist, 124
prior_hessian::CopulaDistImpl::CopulaDist, 154
prior_hessian::GammaDist, 199
prior_hessian::MultivariateNormalDist, 241
prior_hessian::NormalDist, 264
prior_hessian::ParetoDist, 311
prior_hessian::SymmetricBetaDist, 357
prior_hessian::mcmc::MCMCData, 218
rllh_components
prior_hessian::CompositeDist, 125
rllh_const
prior_hessian::AMHCopula, 59
rllh_d2theta_accumulate
prior_hessian::AMHCopula, 60
rllh_dtheta_accumulate
prior_hessian::AMHCopula, 60
rllh_from_iter
prior_hessian::CompositeDist::ComponentDist↩ Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 76
prior_hessian::CompositeDist::ComponentDist↩ Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 87
rllh_grad_accumulate
prior_hessian::AMHCopula, 60
rllh_grad_grad2_accumulate
prior_hessian::AMHCopula, 61
rllh_grad_hess_accumulate
prior_hessian::AMHCopula, 61
RuntimeConvergenceError
prior_hessian::RuntimeConvergenceError, 322
RuntimeSamplingError
prior_hessian::RuntimeSamplingError, 324
RuntimeTypeError
prior_hessian::RuntimeTypeError, 326

sample
prior_hessian::AMHCopula, 61
prior_hessian::CompositeDist, 126, 127
prior_hessian::CopulaDistImpl::CopulaDist, 154
prior_hessian::GammaDist, 199
prior_hessian::MultivariateNormalDist, 242
prior_hessian::NormalDist, 265
prior_hessian::ParetoDist, 312
prior_hessian::ScaledDist, 338
prior_hessian::SymmetricBetaDist, 357
prior_hessian::TruncatedDist, 370
prior_hessian::TruncatedMultivariateDist, 381
prior_hessian::UpperTruncatedDist, 391
prior_hessian::mcmc::MCMCData, 218
scale
prior_hessian::GammaDist, 200
ScaledDist
prior_hessian::ScaledDist, 329, 330
ScaledDist.h, 424
ScaledSymmetricBetaDist
prior_hessian, 14
ScaledSymmetricBetaDist.h, 425
scaling_ratio
prior_hessian::ScaledDist, 343
set_alpha
prior_hessian::ParetoDist, 313
set_beta
prior_hessian::SymmetricBetaDist, 358
set_bounds
prior_hessian::CompositeDist, 127
prior_hessian::CopulaDistImpl::CopulaDist, 155
prior_hessian::ScaledDist, 339
prior_hessian::TruncatedDist, 371
prior_hessian::TruncatedMultivariateDist, 381
prior_hessian::UpperTruncatedDist, 392
set_bounds_from_iter
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
MultivariateDist > >, 76
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
UnivariateDist > >, 87
set_component_names
prior_hessian::CompositeDist, 128
set_copula_theta
prior_hessian::CopulaDistImpl::CopulaDist, 155
set_dim_variables
prior_hessian::CompositeDist, 129
set_lbound
prior_hessian::CompositeDist, 129
prior_hessian::CopulaDistImpl::CopulaDist, 156
prior_hessian::ParetoDist, 314
prior_hessian::ScaledDist, 340
prior_hessian::TruncatedDist, 372
prior_hessian::TruncatedMultivariateDist, 381
prior_hessian::UpperTruncatedDist, 392
set_lbound_from_iter
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
MultivariateDist > >, 77
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist,
UnivariateDist > >, 88
set_min
prior_hessian::ParetoDist, 314
set_mu
prior_hessian::MultivariateNormalDist, 243
prior_hessian::NormalDist, 266
set_param
prior_hessian::GammaDist, 200
prior_hessian::NormalDist, 266
prior_hessian::ParetoDist, 315
prior_hessian::SymmetricBetaDist, 358
set_param_names
prior_hessian::CompositeDist, 130
set_param_value
prior_hessian::CompositeDist, 131
set_params
prior_hessian::AMHCopula, 62
prior_hessian::CompositeDist, 132
prior_hessian::CopulaDistImpl::CopulaDist, 157
prior_hessian::GammaDist, 201, 202
prior_hessian::MultivariateNormalDist, 243, 244
prior_hessian::NormalDist, 267, 268
prior_hessian::ParetoDist, 316
prior_hessian::SymmetricBetaDist, 359, 360
set_params_iter
prior_hessian::AMHCopula, 62
prior_hessian::CopulaDistImpl::CopulaDist, 157
prior_hessian::GammaDist, 202
prior_hessian::MultivariateNormalDist, 245
prior_hessian::NormalDist, 269
prior_hessian::ParetoDist, 317
prior_hessian::SymmetricBetaDist, 360
set_scale
prior_hessian::GammaDist, 202
set_shape
prior_hessian::GammaDist, 203
set_sigma
prior_hessian::MultivariateNormalDist, 245
prior_hessian::NormalDist, 269
set_theta
prior_hessian::AMHCopula, 63
set_ubound
prior_hessian::CompositeDist, 132
prior_hessian::CopulaDistImpl::CopulaDist, 157
prior_hessian::ScaledDist, 340
prior_hessian::TruncatedDist, 373

prior_hessian::TruncatedMultivariateDist, 381
prior_hessian::UpperTruncatedDist, 393
set_ubound_from_iter
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, MultivariateDist > >, 77
prior_hessian::CompositeDist::ComponentDist↩
Adaptor< Dist, meta::EnableIfSubclassT< Dist, UnivariateDist > >, 88
shape
prior_hessian::GammaDist, 203
sigma
prior_hessian::MultivariateNormalDist, 246
prior_hessian::NormalDist, 270
sigma_inv
prior_hessian::MultivariateNormalDist, 246
sqrt2
prior_hessian::constants, 29
sqrt2_inv
prior_hessian::constants, 29
sqrt2pi
prior_hessian::constants, 29
sqrt2pi_inv
prior_hessian::constants, 29
square
prior_hessian, 27
StringVecT
prior_hessian, 15
sum_d10_igen_u
prior_hessian::ArchimedeanCopula::D2Theta_I↩
GenTerms, 165
prior_hessian::ArchimedeanCopula::DTheta_IGen↩
Terms, 184
sum_d20_igen_u
prior_hessian::ArchimedeanCopula::D2Theta_I↩
GenTerms, 165
sum_ieta_01_11_u
prior_hessian::ArchimedeanCopula::D2Theta_I↩
GenTerms, 165
prior_hessian::ArchimedeanCopula::DTheta_IGen↩
Terms, 184
sum_in_order
prior_hessian::meta, 42
sum_ixi_01_u
prior_hessian::ArchimedeanCopula::D2Theta_I↩
GenTerms, 165
sum_log_d1_igen_u
prior_hessian::ArchimedeanCopula::D2Theta_I↩
GenTerms, 165
prior_hessian::ArchimedeanCopula::DTheta_IGen↩
Terms, 185
SymmetricBetaDist
prior_hessian::SymmetricBetaDist, 346
SymmetricBetaDist.cpp, 426
SymmetricBetaDist.h, 427

theta
prior_hessian::AMHCopula, 64
truncated
prior_hessian::TruncatedDist, 374
prior_hessian::TruncatedMultivariateDist, 381
prior_hessian::UpperTruncatedDist, 394
TruncatedDist
prior_hessian::TruncatedDist, 363–365
TruncatedDist.h, 429
TruncatedGammaDist
prior_hessian, 15
TruncatedGammaDist.h, 430
TruncatedMultivariateDist
prior_hessian::TruncatedMultivariateDist, 379
TruncatedMultivariateDist.h, 431
TruncatedMultivariateNormalDist
prior_hessian, 15
TruncatedMultivariateNormalDist.h, 433
TruncatedNormalDist
prior_hessian, 15
TruncatedNormalDist.h, 434
TruncatedParetoDist
prior_hessian, 15
TruncatedParetoDist.h, 435
TypeInfoVecT
prior_hessian, 15

UVecT
prior_hessian, 15
ubound
prior_hessian::CompositeDist, 133
prior_hessian::CopulaDistImpl::CopulaDist, 158
prior_hessian::GammaDist, 204
prior_hessian::MultivariateNormalDist, 247
prior_hessian::NormalDist, 271
prior_hessian::ParetoDist, 318
prior_hessian::ScaledDist, 341
prior_hessian::SymmetricBetaDist, 361
prior_hessian::TruncatedDist, 375
prior_hessian::TruncatedMultivariateDist, 381
prior_hessian::UpperTruncatedDist, 394
unit_normal_cdf
prior_hessian, 27
unit_normal_icdf
prior_hessian, 28
UnivariateDist
prior_hessian::UnivariateDist, 384
UnivariateDist.cpp, 436
UnivariateDist.h, 437
unscaled_lbound
prior_hessian::ScaledDist, 342
unscaled_ubound
prior_hessian::ScaledDist, 342

UpperTruncatedDist
prior_hessian::UpperTruncatedDist, 386, 387
UpperTruncatedDist.h, 438
util.cpp, 439
util.h, 441

value
prior_hessian::meta::all_dists_are_bounded, 46
prior_hessian::meta::is_copula, 211
prior_hessian::meta::is_subclass_of_numeric_template, 214
VecT
prior_hessian, 15

what
prior_hessian::IndexError, 207
prior_hessian::InvalidOperationError, 210
prior_hessian::NotImplementedError, 274
prior_hessian::NumericalOverflowError, 276
prior_hessian::ParameterNameError, 278
prior_hessian::ParameterNameUniquenessError, 280
prior_hessian::ParameterSizeError, 282
prior_hessian::ParameterValueError, 284
prior_hessian::PriorHessianError, 321
prior_hessian::RuntimeConvergenceError, 323
prior_hessian::RuntimeSamplingError, 325
prior_hessian::RuntimeTypeError, 327
what_
prior_hessian::IndexError, 207
prior_hessian::InvalidOperationError, 210
prior_hessian::NotImplementedError, 274
prior_hessian::NumericalOverflowError, 276
prior_hessian::ParameterNameError, 278
prior_hessian::ParameterNameUniquenessError, 280
prior_hessian::ParameterSizeError, 282
prior_hessian::ParameterValueError, 284
prior_hessian::PriorHessianError, 321
prior_hessian::RuntimeConvergenceError, 323
prior_hessian::RuntimeSamplingError, 325
prior_hessian::RuntimeTypeError, 327
what_str
prior_hessian::IndexError, 207
prior_hessian::InvalidOperationError, 210
prior_hessian::NotImplementedError, 274
prior_hessian::NumericalOverflowError, 276
prior_hessian::ParameterNameError, 278
prior_hessian::ParameterNameUniquenessError, 280
prior_hessian::ParameterSizeError, 282
prior_hessian::ParameterValueError, 284
prior_hessian::PriorHessianError, 321
prior_hessian::RuntimeConvergenceError, 323
prior_hessian::RuntimeSamplingError, 325
prior_hessian::RuntimeTypeError, 327

xi_0n_t
prior_hessian::ArchimedeanCopula::D2Theta_GenTerms, 164
xi_n_t
prior_hessian::ArchimedeanCopula::D2_GenTerms, 161