KTAB

Table of Contents

KTAB

This is the top-level README file for KTAB, KAPSARC's open-source toolkit for assembling agent-based models of negotiation and bargaining.

KTAB is written in portable, cross-platform C++11. It was developed and tested on 64bit Windows and Linux platforms. Because no platform-specific code is allowed, porting to Mac or other platforms should be feasible.

KTAB uses CMake for configuration, then your favorite IDE to build. CMake and instructions can be obtained from www.cmake.org. The basic procedure is to configure kutils with CMake, then build the library and examples with your favorite IDE. Then configure kmodel with CMake and build with your favorite IDE. Detailed build instructions can be found in the subdirectories, starting with kutils.

Three examples are provided to illustrate the range of modelling which can be done in the framework and to provide templates for further work. None of them are either calibrated or verified. For rigorous work, one could make whatever local modifications were needed to produce a model which met one's own standards for validation.

KTAB is not a general-purpose agent-based modelling system. All KTAB models represent stochastic decision-making among comparatively small numbers of stakeholder groups (roughly 5 to 50), within the paradigm of "Probabilistic Condorcet Elections". PCEs are used to estimate the likelihood of different outcomes from a collective decision making process, depending on what kinds of options each agent has or can generate, how they value those options, and what kinds of coalitions they can form to support or oppose each option. Examples and details can be found in the documentation.

KTAB is released under The MIT License (Expat). For details, see the following URLs:

- `http://opensource.org/`
- `http://opensource.org/licenses/MIT`

If you are interested in contributing code, ideas, or data to KTAB, please contact `ktab@kapsarc.org`

Namespace Index

Namespace List

Here is a list of all namespaces with brief descriptions:

Hierarchical Index

Class Hierarchy

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

Class Index

Class List

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

File Index

File List

Here is a list of all files with brief descriptions:

Namespace Documentation

DemoLeon Namespace Reference

Classes

- class **LeonActor**
- class **LeonModel**
- class **LeonState**

Functions

- **LeonModel** * **demoSetup** (unsigned int numFctr, unsigned int numCGrp, unsigned int numSect, uint64_t s, **PRNG** *rng)
- void **demoEUEcon** (uint64_t s, unsigned int numF, unsigned int numG, unsigned int numS, **PRNG** *rng)
- void **demoMaxEcon** (uint64_t s, unsigned int numF, unsigned int numG, unsigned int numS, **PRNG** *rng)
- void **demoEUEcon** (uint64_t s, **PRNG** *rng)
- void **demoMaxEcon** (uint64_t s, **PRNG** *rng)

Variables

- const double **TolIFD** = 1E-6

Function Documentation

void DemoLeon::demoEUEcon (uint64_t *s*, PRNG * *rng*)

void DemoLeon::demoEUEcon (uint64_t *s*, unsigned int *numF*, unsigned int *numG*, unsigned int *numS*, PRNG * *rng*)

void DemoLeon::demoMaxEcon (uint64_t *s*, PRNG * *rng*)

void DemoLeon::demoMaxEcon (uint64_t *s*, unsigned int *numF*, unsigned int *numG*, unsigned int *numS*, PRNG * *rng*)

LeonModel * DemoLeon::demoSetup (unsigned int *numFctr*, unsigned int *numCGrp*, unsigned int *numSect*, uint64_t *s*, PRNG * *rng*)

Variable Documentation

const double DemoLeon::TolIFD = 1E-6

DemoMtch Namespace Reference

Classes

- class **MtchActor**
- class **MtchModel**
- class **MtchState**

Functions

- bool **equivMtchPstn** (const **MtchPstn** &mp1, const **MtchPstn** &mp2)
- void **showMtchPstn** (const **MtchPstn** &mp)
- bool **stableMtchState** (unsigned int iter, const **State** *s1)
- void **demoDivideSweets** (uint64_t s, **PRNG** *rng)
- void **demoMaxSupport** (uint64_t s, **PRNG** *rng)
- void **demoMtchSUSN** (uint64_t s, **PRNG** *rng)
- void **multiMtchSUSN** (uint64_t s, **PRNG** *rng)
- bool **oneMtchSUSN** (uint64_t s, **PRNG** *rng)

Function Documentation

void DemoMtch::demoDivideSweets (uint64_t *s*, PRNG * *rng*)

void DemoMtch::demoMaxSupport (uint64_t *s*, PRNG * *rng*)

void DemoMtch::demoMtchSUSN (uint64_t *s*, PRNG * *rng*)

bool DemoMtch::equivMtchPstn (const MtchPstn & *mp1*, const MtchPstn & *mp2*)

void DemoMtch::multiMtchSUSN (uint64_t *s*, PRNG * *rng*)

bool DemoMtch::oneMtchSUSN (uint64_t *s*, PRNG * *rng*)

void DemoMtch::showMtchPstn (const MtchPstn & *mp*)

bool DemoMtch::stableMtchState (unsigned int *iter*, const State * *s1*)

DemoSMP Namespace Reference

Classes

- struct **BargainSMP**
- class **SMPActor**
- class **SMPModel**
- class **SMPState**

Functions

- **SMPModel** * **readCSV** (string fName, **PRNG** *rng)
- void **demoActorUtils** (uint64_t s, **PRNG** *rng)
- void **demoEUSpatial** (unsigned int numA, unsigned int sDim, uint64_t s, **PRNG** *rng)
- void **readEUSpatial** (uint64_t seed, string inputCSV, **PRNG** *rng)

Variables

- const string **appVersion** = "0.1"

Function Documentation

void DemoSMP::demoActorUtils (uint64_t *s*, PRNG * *rng*)

void DemoSMP::demoEUSpatial (unsigned int *numA*, unsigned int *sDim*, uint64_t *s*, PRNG * *rng*)

SMPModel* DemoSMP::readCSV (string *fName*, PRNG * *rng*)

void DemoSMP::readEUSpatial (uint64_t *seed*, string *inputCSV*, PRNG * *rng*)

Variable Documentation

const string DemoSMP::appVersion = "0.1"

DemoWaterMin Namespace Reference

Functions

- double **waterMinProb** (ReportingLevel rl, const **KMatrix** &p0)
- void **minProbErr** ()

Function Documentation

void DemoWaterMin::minProbErr ()

double DemoWaterMin::waterMinProb (ReportingLevel *rl*, const KMatrix & *p0*)

KBase Namespace Reference

Classes

- class **Actor**
- class **GAOpt**
- class **GHCSearch**
- class **KException**
- class **KMatrix**
- class **Model**
- class **MtchGene**
- class **MtchPstn**
- class **Position**
- class **PRNG**
- class **State**
- class **VctrPstn**
- class **VHCSearch**

Enumerations

- enum **VotingRule** : char { **VotingRule::Binary**, **VotingRule::PropBin**, **VotingRule::Proportional**, **VotingRule::PropCbc**, **VotingRule::Cubic** }
- enum **ThirdPartyCommit** { **ThirdPartyCommit::None**, **ThirdPartyCommit::Semi**, **ThirdPartyCommit::Full** }
- enum **ReportingLevel** { **ReportingLevel::Silent**, **ReportingLevel::Low**, **ReportingLevel::Medium**, **ReportingLevel::High**, **ReportingLevel::Debugging** }

Functions

- string **vrName** (**VotingRule** vr)
- string **tpcName** (**ThirdPartyCommit** tpc)
- unsigned int **crossSite** (**PRNG** *rng, unsigned int nc)
- **KMatrix trans** (const **KMatrix** &m1)
- double **norm** (const **KMatrix** &m1)
- double **sum** (const **KMatrix** &m1)
- double **mean** (const **KMatrix** &m1)
- double **stdv** (const **KMatrix** &m1)
- double **maxAbs** (const **KMatrix** &m)
- tuple< unsigned int, unsigned int > **ndxMaxAbs** (const **KMatrix** &m)
- double **lCorr** (const **KMatrix** &m1, const **KMatrix** &m2)
- double **dot** (const **KMatrix** &m1, const **KMatrix** &m2)
- **KMatrix operator+** (const **KMatrix** &m1, double x)
- **KMatrix operator-** (const **KMatrix** &m1, double x)
- bool **sameShape** (const **KMatrix** &m1, const **KMatrix** &m2)
- **KMatrix operator+** (const **KMatrix** &m1, const **KMatrix** &m2)
- **KMatrix operator-** (const **KMatrix** &m1, const **KMatrix** &m2)
- **KMatrix operator*** (double x, const **KMatrix** &m1)
- **KMatrix operator/** (const **KMatrix** &m1, double x)
- **KMatrix operator*** (const **KMatrix** &m1, const **KMatrix** &m2)
- **KMatrix inv** (const **KMatrix** &m)
- **KMatrix iMat** (unsigned int n)
- **KMatrix makePerp** (const **KMatrix** &x, const **KMatrix** &p)
- **KMatrix joinH** (const **KMatrix** &mL, const **KMatrix** &mR)

- **KMatrix joinV** (const **KMatrix** &mT, const **KMatrix** &mB)
- std::chrono::time_point< std::chrono::system_clock, std::chrono::system_clock::duration > **displayProgramStart** ()
- void **displayProgramEnd** (std::chrono::time_point< std::chrono::system_clock, std::chrono::system_clock::duration > &sTime)
- double **rescale** (double x, double x0, double x1, double y0, double y1)
- template<typename T > T **popBack** (vector< T > &v)
- uint64_t **qTrans** (uint64_t s)
- **KMatrix projPos** (const **KMatrix** &w)
- **KMatrix projBox** (const **KMatrix** &lb, const **KMatrix** &ub, const **KMatrix** &w)
- tuple< **KMatrix**, unsigned int, **KMatrix** > **viABG** (const **KMatrix** &xInit, function< **KMatrix**(const **KMatrix** &x)> F, function< **KMatrix**(const **KMatrix** &x)> P, double beta, double thresh, unsigned int iMax)
- tuple< **KMatrix**, unsigned int, **KMatrix** > **solveLVI_BSHe96** (const **KMatrix** &M, const **KMatrix** &q, function< **KMatrix**(const **KMatrix** &)> pK, **KMatrix** u0, const double eps, const unsigned int iMax)
- tuple< **KMatrix**, **KMatrix**, **KMatrix**, **KMatrix** > **antiLemke** (unsigned int n)

Enumeration Type Documentation

enum KBase::ReportingLevel`[strong]`

Enumerator

Silent
Low
Medium
High
Debugging

enum KBase::ThirdPartyCommit`[strong]`

Enumerator

None
Semi
Full

enum KBase::VotingRule : char`[strong]`

Enumerator

Binary
PropBin
Proportional
PropCbc
Cubic

Function Documentation

tuple<KMatrix, KMatrix, KMatrix, KMatrix> KBase::antiLemke (unsigned int ***n*****)**

unsigned int KBase::crossSite (PRNG * ***rng*****, unsigned int** ***nc*****)**

void KBase::displayProgramEnd (std::chrono::time_point< std::chrono::system_clock, std::chrono::system_clock::duration > & ***sTime*****)**

std::chrono::time_point< std::chrono::system_clock, std::chrono::system_clock::duration > KBase::displayProgramStart ()

double KBase::dot (const KMatrix & ***m1*****, const KMatrix &** ***m2*****)**

KMatrix KBase::iMat (unsigned int ***n*****)**

KMatrix KBase::inv (const KMatrix & ***m*****)**

KMatrix KBase::joinH (const KMatrix & ***mL*****, const KMatrix &** ***mR*****)**

KMatrix KBase::joinV (const KMatrix & ***mT*****, const KMatrix &** ***mB*****)**

double KBase::lCorr (const KMatrix & ***m1*****, const KMatrix &** ***m2*****)**

KMatrix KBase::makePerp (const KMatrix & ***x*****, const KMatrix &** ***p*****)**

double KBase::maxAbs (const KMatrix & ***m*****)**

double KBase::mean (const KMatrix & ***m1*****)**

tuple< unsigned int, unsigned int > KBase::ndxMaxAbs (const KMatrix & ***m*****)**

double KBase::norm (const KMatrix & ***m1*****)**

KMatrix KBase::operator* (double ***x*****, const KMatrix &** ***m1*****)**

KMatrix KBase::operator* (const KMatrix & ***m1*****, const KMatrix &** ***m2*****)**

KMatrix KBase::operator+ (const KMatrix & ***m1*****, double** ***x*****)**

KMatrix KBase::operator+ (const KMatrix & ***m1*****, const KMatrix &** ***m2*****)**

KMatrix KBase::operator- (const KMatrix & ***m1*****, double** ***x*****)**

KMatrix KBase::operator- (const KMatrix & ***m1*****, const KMatrix &** ***m2*****)**

KMatrix KBase::operator/ (const KMatrix & ***m1*****, double** ***x*****)**

template<typename T > T KBase::popBack (vector< T > & ***v*****)**

KMatrix KBase::projBox (const KMatrix & *lb*, const KMatrix & *ub*, const KMatrix & *w*)

KMatrix KBase::projPos (const KMatrix & *w*)

uint64_t KBase::qTrans (uint64_t *s*)

double KBase::rescale (double *x*, double *x0*, double *x1*, double *y0*, double *y1*)

bool KBase::sameShape (const KMatrix & *m1*, const KMatrix & *m2*)

tuple< KMatrix, unsigned int, KMatrix > KBase::solveLVI_BSHe96 (const KMatrix & *M*, const KMatrix & *q*, function< KMatrix(const KMatrix &)> *pK*, KMatrix *u0*, const double *eps*, const unsigned int *iMax*)

double KBase::stdv (const KMatrix & *m1*)

double KBase::sum (const KMatrix & *m1*)

string KBase::tpcName (ThirdPartyCommit *tpc*)

KMatrix KBase::trans (const KMatrix & *m1*)

tuple< KMatrix, unsigned int, KMatrix > KBase::viABG (const KMatrix & *xInit*, function< KMatrix(const KMatrix &x)> *F*, function< KMatrix(const KMatrix &x)> *P*, double *beta*, double *thresh*, unsigned int *iMax*)

string KBase::vrName (VotingRule *vr*)

MDemo Namespace Reference

Classes

- class **ZActor**

Functions

- void **demoPCE** (uint64_t s, **PRNG** *rng)
- void **demoSpVSR** (uint64_t s, **PRNG** *rng)

Function Documentation

void MDemo::demoPCE (uint64_t *s*, PRNG * *rng*)

void MDemo::demoSpVSR (uint64_t *s*, PRNG * *rng*)

UDemo Namespace Reference

Classes

- class **TargetedBV**

Typedefs

- typedef vector< bool > **BVec**

Functions

- void **show** (string str, const **KMatrix** &m, string fs)
- double **nProd** (double x, double y)
- double **bsu** (double d, double R)
- double **bvu** (const **KBase::KMatrix** &d, const **KBase::KMatrix** &s, double R)
- void **demoMatrix** (**PRNG** *rng)
- void **demoABG00** (**PRNG** *rng)
- double **eNorm** (const **KMatrix** &a, const **KMatrix** &x)
- **KMatrix** **eUnitize** (const **KMatrix** &a, const **KMatrix** &x)
- **KMatrix** **projEllipse** (const **KMatrix** &a, const **KMatrix** &w)
- void **demoEllipseLVI** (**PRNG** *rng, unsigned int n)
- tuple< **KMatrix**, **KMatrix**, **KMatrix**, **KMatrix** > **antiLemke** (unsigned int n)
- void **demoAntiLemke** (**PRNG** *rng, unsigned int n)
- void **demoEllipse** (**PRNG** *rng)
- void **demoGA** (**PRNG** *rng)
- void **demoGHC** (**PRNG** *rng)
- void **demoVHC00** (**PRNG** *rng)
- void **demoVHC01** (**PRNG** *rng)
- void **demoVHC02** (**PRNG** *rng)
- void **demoVHC03** (**PRNG** *rng)
- void **parallelMatrixMult** (**PRNG** *rng)

Typedef Documentation

typedef vector<bool> UDemo::BVec

Function Documentation

tuple<KMatrix, KMatrix, KMatrix, KMatrix> UDemo::antiLemke (unsigned int ***n*****)**

double UDemo::bsu (double ***d*****, double** ***R*****)**

double UDemo::bvu (const KBase::KMatrix & ***d*****, const KBase::KMatrix &** ***s*****, double** ***R*****)**

void UDemo::demoABG00 (PRNG * ***rng*****)**

void UDemo::demoAntiLemke (PRNG * ***rng*****, unsigned int** ***n*****)**

void UDemo::demoEllipse (PRNG * ***rng*****)**

void UDemo::demoEllipseLVI (PRNG * ***rng*****, unsigned int** ***n*****)**

void UDemo::demoGA (PRNG * ***rng*****)**

void UDemo::demoGHC (PRNG * ***rng*****)**

void UDemo::demoMatrix (PRNG * ***rng*****)**

void UDemo::demoVHC00 (PRNG * ***rng*****)**

void UDemo::demoVHC01 (PRNG * ***rng*****)**

void UDemo::demoVHC02 (PRNG * ***rng*****)**

void UDemo::demoVHC03 (PRNG * ***rng*****)**

double UDemo::eNorm (const KMatrix & ***a*****, const KMatrix &** ***x*****)**

KMatrix UDemo::eUnitize (const KMatrix & ***a*****, const KMatrix &** ***x*****)**

double UDemo::nProd (double ***x*****, double** ***y*****)**

void UDemo::parallelMatrixMult (PRNG * ***rng*****)**

KMatrix UDemo::projEllipse (const KMatrix & ***a*****, const KMatrix &** ***w*****)**

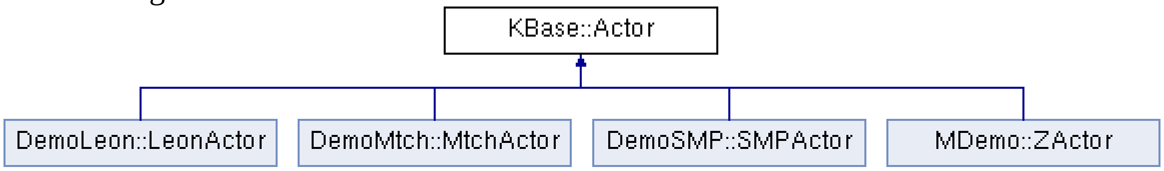
void UDemo::show (string ***str*****, const KMatrix &** ***m*****, string** ***fs*****)**

Class Documentation

KBase::Actor Class Reference

`#include <kmodel.h>`

Inheritance diagram for KBase::Actor:



Public Member Functions

- **Actor** (string n, string d)
- virtual **~Actor** ()
- virtual double **vote** (unsigned int p1, unsigned int p2, const **State** *st) const =0

Static Public Member Functions

- static double **thirdPartyVoteSU** (double wk, **VotingRule** vr, **ThirdPartyCommit** comm, double pik, double pjk, double uki, double ukj, double ukk)
- static double **vProbLittle** (**VotingRule** vr, double wn, double uni, double unj, double contrib_i_ij, double contrib_j_ij)

Public Attributes

- string **name**
- string **desc**

Constructor & Destructor Documentation

KBase::Actor::Actor (string *n*, string *d*)

KBase::Actor::~Actor () `[virtual]`

Member Function Documentation

double KBase::Actor::thirdPartyVoteSU (double *wk*, VotingRule *vr*, ThirdPartyCommit *comm*, double *pik*, double *pjk*, double *uki*, double *ukj*, double *ukk*) `[static]`

virtual double KBase::Actor::vote (unsigned int *p1*, unsigned int *p2*, const State * *st*) const `[pure virtual]`

Implemented in **DemoSMP::SMPActor** (*p.52*), **DemoMtch::MtchActor** (*p.39*), **DemoLeon::LeonActor** (*p.29*), and **MDemo::ZActor** (*p.65*).

double KBase::Actor::vProbLittle (VotingRule *vr*, double *wn*, double *uni*, double *unj*, double *contrib_i_ij*, double *contrib_j_ij*)`[static]`

Member Data Documentation

string KBase::Actor::desc

string KBase::Actor::name

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

- **kmodel.h**
- **kmodel.cpp**

DemoSMP::BargainSMP Struct Reference

```
#include <demosmp.h>
```

Public Member Functions

- **BargainSMP** (const **SMPActor** *ai, const **SMPActor** *ar, const **VctrPstn** &pi, const **VctrPstn** &pr)
- **~BargainSMP** ()

Public Attributes

- const **SMPActor** * **actInit**
- const **SMPActor** * **actRcvr**
- **VctrPstn** **posInit**
- **VctrPstn** **posRcvr**

Constructor & Destructor Documentation

DemoSMP::BargainSMP::BargainSMP (const SMPActor * *ai*, const SMPActor * *ar*, const VctrPstn & *pi*, const VctrPstn & *pr*)

DemoSMP::BargainSMP::~BargainSMP ()

Member Data Documentation

const SMPActor* DemoSMP::BargainSMP::actInit

const SMPActor* DemoSMP::BargainSMP::actRcvr

VctrPstn DemoSMP::BargainSMP::posInit

VctrPstn DemoSMP::BargainSMP::posRcvr

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

- **demosmp.h**
- **demosmp.cpp**

KBase::GAOpt< GAP > Class Template Reference

```
#include <gaopt.h>
```

Public Member Functions

- **GAOpt** (unsigned int s)
- virtual **~GAOpt** ()
- void **init** (vector< GAP * > ipop)
- void **fill** (**PRNG** ***rng**)
- void **run** (**PRNG** ***rng**, double c, double m, unsigned int maxI, double sTh, unsigned int maxS, **ReportingLevel** srl, unsigned int &iter, unsigned int &sIter)
- tuple< double, GAP * > **getNth** (unsigned int n)
- void **show** ()
- void **sortPop** ()

Public Attributes

- function< tuple< GAP *, GAP * >const GAP *g1, const GAP *g2, **PRNG** ***rng**)> **cross**
- function< GAP *(const GAP *g1, **PRNG** ***rng**)> **mutate**
- function< double(const GAP *g1)> **eval**
- function< void(const GAP *)> **showGene**
- function< GAP *(**PRNG** ***rng**)> **makeGene**
- function< bool(const GAP *g1, const GAP *g2)> **equiv**

Protected Member Functions

- void **step** ()
- void **mutatePop** ()
- void **crossPop** ()
- void **dropDups** ()
- void **selectPop** ()
- GAP * **mutateOne** (const GAP *g1, **PRNG** ***rng**)
- tuple< GAP *, GAP * > **crossPair** (const GAP *g1, const GAP *g2, **PRNG** ***rng**)
- void **cyclicApply** (function< void(unsigned int i)> fn, double f)

Protected Attributes

- vector< tuple< double, GAP * > > **gpool**
- unsigned int **pSize**
- double **cFrac**
- double **mFrac**
- **PRNG** * **rng**

Constructor & Destructor Documentation

template<class GAP > KBase::GAOpt< GAP >::GAOpt (unsigned int *s*)

template<class GAP > KBase::GAOpt< GAP >::~GAOpt () `[virtual]`

Member Function Documentation

template<class GAP > tuple<GAP*, GAP*> KBase::GAOpt< GAP >::crossPair (const GAP * *g1*, const GAP * *g2*, PRNG * *rng*) `[protected]`

template<class GAP > void KBase::GAOpt< GAP >::crossPop () `[protected]`

template<class GAP > void KBase::GAOpt< GAP >::cyclicApply (function< void(unsigned int i)> *fn*, double *f*) `[protected]`

template<class GAP > void KBase::GAOpt< GAP >::dropDups () `[protected]`

template<class GAP > void KBase::GAOpt< GAP >::fill (PRNG * *rng*)

template<class GAP > tuple< double, GAP * > KBase::GAOpt< GAP >::getNth (unsigned int *n*)

template<class GAP > void KBase::GAOpt< GAP >::init (vector< GAP * > *ipop*)

template<class GAP > GAP* KBase::GAOpt< GAP >::mutateOne (const GAP * *g1*, PRNG * *rng*) `[protected]`

template<class GAP > void KBase::GAOpt< GAP >::mutatePop () `[protected]`

template<class GAP > void KBase::GAOpt< GAP >::run (PRNG * *rng*, double *c*, double *m*, unsigned int *maxI*, double *sTh*, unsigned int *maxS*, ReportingLevel *srl*, unsigned int & *iter*, unsigned int & *sIter*)

template<class GAP > void KBase::GAOpt< GAP >::selectPop () `[protected]`

template<class GAP > void KBase::GAOpt< GAP >::show ()

template<class GAP > void KBase::GAOpt< GAP >::sortPop ()

template<class GAP > void KBase::GAOpt< GAP >::step () `[protected]`

Member Data Documentation

template<class GAP > double KBase::GAOpt< GAP >::cFrac `[protected]`

template<class GAP > function<tuple<GAP*, GAP*>const GAP* g1, const GAP* g2, PRNG* rng)> KBase::GAOpt< GAP >::cross

template<class GAP > function<bool(const GAP* g1, const GAP* g2)> KBase::GAOpt< GAP >::equiv

template<class GAP > function<double(const GAP* g1)> KBase::GAOpt< GAP >::eval

template<class GAP > vector< tuple<double, GAP* > > KBase::GAOpt< GAP >::gpool `[protected]`

template<class GAP > function<GAP* (PRNG* rng)> KBase::GAOpt< GAP >::makeGene

template<class GAP > double KBase::GAOpt< GAP >::mFrac `[protected]`

template<class GAP > function<GAP* (const GAP* g1, PRNG* rng)> KBase::GAOpt< GAP >::mutate

template<class GAP > unsigned int KBase::GAOpt< GAP >::pSize `[protected]`

template<class GAP > PRNG* KBase::GAOpt< GAP >::rng `[protected]`

template<class GAP > function<void(const GAP*)> KBase::GAOpt< GAP >::showGene

---

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

- **gaopt.h**

KBase::GHCSearch< HCP > Class Template Reference

```
#include <hcsearch.h>
```

Public Member Functions

- **GHCSearch** ()
- virtual **~GHCSearch** ()
- tuple< double, HCP, unsigned int, unsigned int > **run** (HCP p0, **ReportingLevel** srl, unsigned int iMax, unsigned int sMax, double sTol)

Public Attributes

- function< double(const HCP)> **eval**
- function< vector< HCP >const HCP)> **nghbrs**
- function< void(const HCP)> **show**

Constructor & Destructor Documentation

template<class HCP > KBase::GHCSearch< HCP >::GHCSearch ()

template<class HCP > KBase::GHCSearch< HCP >::~GHCSearch () `[virtual]`

Member Function Documentation

template<class HCP > tuple< double, HCP, unsigned int, unsigned int > KBase::GHCSearch< HCP >::run (HCP *p0*, ReportingLevel *srl*, unsigned int *iMax*, unsigned int *sMax*, double *sTol*)

Member Data Documentation

template<class HCP> function<double(const HCP)> KBase::GHCSearch< HCP >::eval

template<class HCP> function<vector<HCP>const HCP)> KBase::GHCSearch< HCP >::nghbrs

template<class HCP> function<void(const HCP)> KBase::GHCSearch< HCP >::show

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

- **hcsearch.h**

KBase::KException Class Reference

`#include <kutils.h>`

Public Member Functions

- **KException** (string m)
- virtual **~KException** ()

Public Attributes

- string **msg**

Constructor & Destructor Documentation

KBase::KException::KException (string *m*)

KBase::KException::~KException () `[virtual]`

Member Data Documentation

string KBase::KException::msg

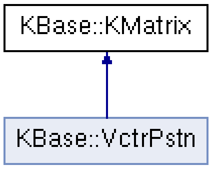
The documentation for this class was generated from the following files:

- **kutils.h**
- **kutils.cpp**

KBase::KMatrix Class Reference

```
#include <kmatrix.h>
```

Inheritance diagram for KBase::KMatrix:



Public Member Functions

- **KMatrix** ()
- **KMatrix** (unsigned int nr, unsigned int nc)
- double **operator()** (unsigned int i, unsigned int j) const
- double & **operator()** (unsigned int i, unsigned int j)
- void **printf** (string) const
- unsigned int **numR** () const
- unsigned int **numC** () const
- vector< double >::iterator **begin** ()
- vector< double >::iterator **end** ()
- vector< double >::const_iterator **cbegin** ()
- vector< double >::const_iterator **cend** ()
- vector< double >::const_iterator **begin** () const
- vector< double >::const_iterator **end** () const

Static Public Member Functions

- static **KMatrix uniform** (**PRNG** *rng, unsigned int nr, unsigned int nc, double a, double b)
- static **KMatrix map** (function< double(unsigned int i, unsigned int j)> f, unsigned int nr, unsigned int nc)
- static void **mapV** (function< void(unsigned int i, unsigned int j)> f, unsigned int nr, unsigned int nc)
- static **KMatrix arrayInit** (const double mv[], const unsigned int &rows, const unsigned int &clms)

Friends

- **KMatrix inv** (const **KMatrix** &m)

Constructor & Destructor Documentation

KBase::KMatrix::KMatrix ()

KBase::KMatrix::KMatrix (unsigned int *nr*, unsigned int *nc*)

Member Function Documentation

KMatrix KBase::KMatrix::arrayInit (const double *mv*[], const unsigned int & *rows*, const unsigned int & *clms*)`[static]`

vector<double>::iterator KBase::KMatrix::begin ()`[inline]`

vector<double>::const_iterator KBase::KMatrix::begin () const`[inline]`

vector<double>::const_iterator KBase::KMatrix::cbegin ()`[inline]`

vector<double>::const_iterator KBase::KMatrix::cend ()`[inline]`

vector<double>::iterator KBase::KMatrix::end ()`[inline]`

vector<double>::const_iterator KBase::KMatrix::end () const`[inline]`

KMatrix KBase::KMatrix::map (function< double(unsigned int i, unsigned int j)> *f*, unsigned int *nr*, unsigned int *nc*)`[static]`

void KBase::KMatrix::mapV (function< void(unsigned int i, unsigned int j)> *f*, unsigned int *nr*, unsigned int *nc*)`[static]`

unsigned int KBase::KMatrix::numC () const

unsigned int KBase::KMatrix::numR () const

double KBase::KMatrix::operator() (unsigned int *i*, unsigned int *j*) const

double & KBase::KMatrix::operator() (unsigned int *i*, unsigned int *j*)

void KBase::KMatrix::printf (string *fs*) const

KMatrix KBase::KMatrix::uniform (PRNG * *rng*, unsigned int *nr*, unsigned int *nc*, double *a*, double *b*)`[static]`

Friends And Related Function Documentation

KMatrix inv (const KMatrix & *m*)`[friend]`

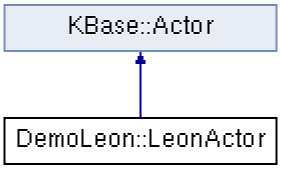
The documentation for this class was generated from the following files:

- **kmatrix.h**
- **kmatrix.cpp**

DemoLeon::LeonActor Class Reference

```
#include <demoleon.h>
```

Inheritance diagram for DemoLeon::LeonActor:



Public Member Functions

- **LeonActor** (string n, string d, **LeonModel** *em, unsigned int id)
- **~LeonActor** ()
- double **vote** (unsigned int p1, unsigned int p2, const **State** *st) const
- virtual double **vote** (const **Position** *ap1, const **Position** *ap2) const
- double **posUtil** (const **Position** *ap1) const
- void **randomize** (**PRNG** *rng)
- void **setShareUtilScale** (const **KMatrix** &runs)
- double **shareToUtil** (double gdpShare) const

Public Attributes

- **LeonModel** * **eMod**
- unsigned int **idNum**
- **KMatrix vCap**
- VotingRule **vr**
- double **minS**
- double **refS**
- double **refU**
- double **maxS**

Additional Inherited Members

Constructor & Destructor Documentation

DemoLeon::LeonActor::LeonActor (string ***n*****, string** ***d*****, LeonModel *** ***em*****, unsigned int** ***id*****)**

DemoLeon::LeonActor::~LeonActor ()

Member Function Documentation

double DemoLeon::LeonActor::posUtil (const Position * *ap1*) const

void DemoLeon::LeonActor::randomize (PRNG * *rng*)

void DemoLeon::LeonActor::setShareUtilScale (const KMatrix & *runs*)

double DemoLeon::LeonActor::shareToUtil (double *gdpShare*) const

double DemoLeon::LeonActor::vote (unsigned int *p1*, unsigned int *p2*, const State * *st*) const[virtual]

Implements **KBase::Actor** (*p.18*).

double DemoLeon::LeonActor::vote (const Position * *ap1*, const Position * *ap2*) const[virtual]

Member Data Documentation

LeonModel* DemoLeon::LeonActor::eMod

unsigned int DemoLeon::LeonActor::idNum

double DemoLeon::LeonActor::maxS

double DemoLeon::LeonActor::minS

double DemoLeon::LeonActor::refS

double DemoLeon::LeonActor::refU

KMatrix DemoLeon::LeonActor::vCap

VotingRule DemoLeon::LeonActor::vr

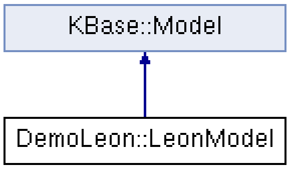
The documentation for this class was generated from the following files:

- **demoleon.h**
- **demoleon.cpp**

DemoLeon::LeonModel Class Reference

```
#include <demoleon.h>
```

Inheritance diagram for DemoLeon::LeonModel:



Public Member Functions

- **LeonModel** (**PRNG** *r)
- virtual **~LeonModel** ()
- tuple< **KMatrix**, **KMatrix**, **KMatrix**, **KMatrix** > **makeBaseYear** (unsigned int numF, unsigned int numCG, unsigned int numS, **PRNG *rng**)
- void **makeIOModel** (const **KMatrix** &trns, const **KMatrix** &rev, const **KMatrix** &xprt, const **KMatrix** &cons, **PRNG** ***rng**)
- **KMatrix xprtDemand** (const **KBase::KMatrix** &tau) const
- **KMatrix randomFTax** (**PRNG** ***rng**)
- **KMatrix makeFTax** (const **KBase::KMatrix** &tax) const
- double **infsDegree** (const **KMatrix** &tax) const
- **KMatrix vaShares** (const **KMatrix** &tax, bool normalizeSharesP) const
- **KMatrix monteCarloShares** (unsigned int nRuns, **KBase::PRNG** ***rng**)

Static Public Member Functions

- static double **stateDist** (const **LeonState** *s1, const **LeonState** *s2)

Protected Attributes

- unsigned int **L**
- unsigned int **M**
- unsigned int **N**
- double **maxSub**
- double **maxTax**
- **KMatrix x0**
- **KMatrix eps**
- **KMatrix aL**
- **KMatrix bL**
- **KMatrix rho**
- **KMatrix vas**

Friends

- class **LeonActor**

Additional Inherited Members

Constructor & Destructor Documentation

DemoLeon::LeonModel::LeonModel (PRNG *   *r*)

DemoLeon::LeonModel::~LeonModel ()`[virtual]`

Member Function Documentation

double DemoLeon::LeonModel::infsDegree (const KMatrix &   *tax*) const

tuple< KMatrix, KMatrix, KMatrix, KMatrix > DemoLeon::LeonModel::makeBaseYear (unsigned int *numF*, unsigned int   *numCG*, unsigned int   *numS*, PRNG *   *rng*)

KMatrix DemoLeon::LeonModel::makeFTax (const KBase::KMatrix &   *tax*) const

void DemoLeon::LeonModel::makeIOModel (const KMatrix &   *trns*, const KMatrix &   *rev*, const KMatrix &   *xprt*, const KMatrix &   *cons*, PRNG *   *rng*)

KMatrix DemoLeon::LeonModel::monteCarloShares (unsigned int   *nRuns*, KBase::PRNG *   *rng*)

KMatrix DemoLeon::LeonModel::randomFTax (PRNG *   *rng*)

double DemoLeon::LeonModel::stateDist (const LeonState *   *s1*, const LeonState * *s2*)`[static]`

KMatrix DemoLeon::LeonModel::vaShares (const KMatrix &   *tax*, bool   *normalizeSharesP*) const

KMatrix DemoLeon::LeonModel::xprtDemand (const KBase::KMatrix &   *tau*) const

Friends And Related Function Documentation

friend class LeonActor`[friend]`

Member Data Documentation

KMatrix DemoLeon::LeonModel::aL `[protected]`

KMatrix DemoLeon::LeonModel::bL `[protected]`

KMatrix DemoLeon::LeonModel::eps `[protected]`

unsigned int DemoLeon::LeonModel::L `[protected]`

unsigned int DemoLeon::LeonModel::M `[protected]`

double DemoLeon::LeonModel::maxSub `[protected]`

double DemoLeon::LeonModel::maxTax `[protected]`

unsigned int DemoLeon::LeonModel::N `[protected]`

KMatrix DemoLeon::LeonModel::rho `[protected]`

KMatrix DemoLeon::LeonModel::vas `[protected]`

KMatrix DemoLeon::LeonModel::x0 `[protected]`

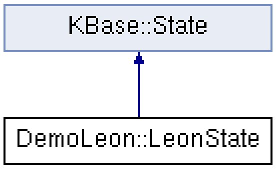
The documentation for this class was generated from the following files:

- **demoleon.h**
- **demoleon.cpp**

DemoLeon::LeonState Class Reference

`#include <demoleon.h>`

Inheritance diagram for DemoLeon::LeonState:



Public Member Functions

- **LeonState** (**LeonModel** *em)
- **~LeonState** ()
- virtual **KMatrix pDist** (int persp) const
- virtual void **setAUtil** (ReportingLevel rl)
- **LeonState** * **stepSUSN** ()

Public Attributes

- const **LeonModel** * **eMod**

Protected Member Functions

- **LeonState** * **doSUSN** (ReportingLevel rl) const

Constructor & Destructor Documentation

DemoLeon::LeonState::LeonState (LeonModel * *em*)

DemoLeon::LeonState::~LeonState ()

Member Function Documentation

LeonState * DemoLeon::LeonState::doSUSN (ReportingLevel *rl*) const `[protected]`

KMatrix DemoLeon::LeonState::pDist (int *persp*) const `[virtual]`

Implements **KBase::State** (*p.58*).

void DemoLeon::LeonState::setAUtil (ReportingLevel *rl*) `[virtual]`

LeonState * DemoLeon::LeonState::stepSUSN ()

Member Data Documentation

const LeonModel* DemoLeon::LeonState::eMod

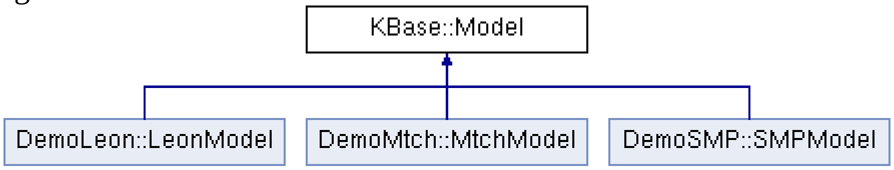
The documentation for this class was generated from the following files:

- **demoleon.h**
- **demoleon.cpp**

KBase::Model Class Reference

`#include <kmodel.h>`

Inheritance diagram for KBase::Model:



Public Types

- enum **VPModel** { **VPModel::Linear**, **VPModel::Square** }

Public Member Functions

- **Model** (**PRNG** *r)
- virtual **~Model** ()
- void **run** ()
- virtual unsigned int **addActor** (**Actor** *a)
- int **actrNdx** (const **Actor** *a) const
- int **addState** (**State** *s)

Static Public Member Functions

- static double **nProd** (double x, double y)
- static double **vote** (**VotingRule** vr, double wi, double uij, double uik)
- static **KMatrix coalitions** (function< double(unsigned int ak, unsigned int pi, unsigned int pj)> vfn, unsigned int **numAct**, unsigned int numOpt)
- static string **VPMName** (**VPModel** vpm)
- static **KMatrix vProb** (**VPModel** vpm, const **KMatrix** &c)
- static **KMatrix vProb** (**VotingRule** vr, **VPModel** vpm, const **KMatrix** &w, const **KMatrix** &u)
- static **KMatrix probCE** (const **KMatrix** &pv)
- static **KMatrix markovPCE** (const **KMatrix** &pv)
- static **KMatrix condPCE** (const **KMatrix** &pv)
- static **KMatrix scalarPCE** (unsigned int **numAct**, unsigned int numOpt, const **KMatrix** &w, const **KMatrix** &u, **VotingRule** vr, **VPModel** vpm, **ReportingLevel** rl)

Public Attributes

- function< bool(unsigned int iter, const **State** *s)> **stop**
- vector< **Actor** * > **actrs**
- unsigned int **numAct**
- **PRNG** * **rng**
- vector< **State** * > **history**

Member Enumeration Documentation

enum KBase::Model::VPModel `[strong]`

Enumerator

Linear
Square

Constructor & Destructor Documentation

KBase::Model::Model (PRNG * *r*)

KBase::Model::~Model ()`[virtual]`

Member Function Documentation

int KBase::Model::actrNdx (const Actor * *a*) const

unsigned int KBase::Model::addActor (Actor * *a*)`[virtual]`

int KBase::Model::addState (State * *s*)

KMatrix KBase::Model::coalitions (function< double(unsigned int ak, unsigned int pi, unsigned int pj)> *vfn*, unsigned int *numAct*, unsigned int *numOpt*)`[static]`

KMatrix KBase::Model::condPCE (const KMatrix & *pv*)`[static]`

KMatrix KBase::Model::markovPCE (const KMatrix & *pv*)`[static]`

double KBase::Model::nProd (double *x*, double *y*)`[static]`

KMatrix KBase::Model::probCE (const KMatrix & *pv*)`[static]`

void KBase::Model::run ()

KMatrix KBase::Model::scalarPCE (unsigned int *numAct*, unsigned int *numOpt*, const KMatrix & *w*, const KMatrix & *u*, VotingRule *vr*, VPModel *vpm*, ReportingLevel *rl*)`[static]`

double KBase::Model::vote (VotingRule *vr*, double *wi*, double *uij*, double *uik*)`[static]`

string KBase::Model::VPMName (VPModel *vpm*)`[static]`

KMatrix KBase::Model::vProb (VPModel *vpm*, const KMatrix & *c*)`[static]`

KMatrix KBase::Model::vProb (VotingRule *vr*, VPModel *vpm*, const KMatrix & *w*, const KMatrix & *u*)`[static]`

Member Data Documentation

vector<Actor*> KBase::Model::actrs

vector<State*> KBase::Model::history

unsigned int KBase::Model::numAct

PRNG* KBase::Model::rng

function<bool(unsigned int iter, const State* s)> KBase::Model::stop

---

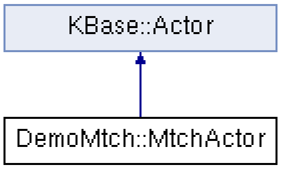
The documentation for this class was generated from the following files:

- **kmodel.h**
- **kmodel.cpp**

DemoMtch::MtchActor Class Reference

```
#include <demomtch.h>
```

Inheritance diagram for DemoMtch::MtchActor:



Public Types

- enum **PropModel** { **PropModel::ExpUtil**, **PropModel::Probability**, **PropModel::AgreeUtil** }

Public Member Functions

- **MtchActor** (string n, string d)
- **~MtchActor** ()
- double **vote** (unsigned int p1, unsigned int p2, const **State** *st) const
- virtual double **vote** (const **Position** *ap1, const **Position** *ap2) const
- double **posUtil** (const **Position** *ap1) const
- void **randomize** (**PRNG** *rng, double minCap, double maxCap, unsigned int id, unsigned int numI)
- tuple< double, **MtchPstn** > **maxProbEUPstn** (**PropModel** pm, const **MtchState** *mst) const

Static Public Member Functions

- static **MtchPstn** * **rPos** (unsigned int numI, unsigned int numA, **PRNG** *rng)
- static **MtchActor** * **rAct** (unsigned int numI, double minCap, double maxCap, **PRNG** *rng, unsigned int i)

Public Attributes

- unsigned int **idNum**
- VotingRule **vr**
- **PropModel pMod**
- double **sCap**
- vector< double > **vals**

Member Enumeration Documentation

enum DemoMtch::MtchActor::PropModel `[strong]`

Enumerator

ExpUtil
Probability
AgreeUtil

Constructor & Destructor Documentation

DemoMtch::MtchActor::MtchActor (string *n*, string *d*)

DemoMtch::MtchActor::~MtchActor ()

Member Function Documentation

tuple< double, MtchPstn > DemoMtch::MtchActor::maxProbEUPstn (PropModel *pm*, const MtchState * *mst*) const

double DemoMtch::MtchActor::posUtil (const Position * *ap1*) const

MtchActor * DemoMtch::MtchActor::rAct (unsigned int *numI*, double *minCap*, double *maxCap*, PRNG * *rng*, unsigned int *i*)`[static]`

void DemoMtch::MtchActor::randomize (PRNG * *rng*, double *minCap*, double *maxCap*, unsigned int *id*, unsigned int *numI*)

MtchPstn * DemoMtch::MtchActor::rPos (unsigned int *numI*, unsigned int *numA*, PRNG * *rng*)`[static]`

double DemoMtch::MtchActor::vote (unsigned int *p1*, unsigned int *p2*, const State * *st*) const`[virtual]`

Implements **KBase::Actor** (*p.18*).

double DemoMtch::MtchActor::vote (const Position * *ap1*, const Position * *ap2*) const`[virtual]`

Member Data Documentation

unsigned int DemoMtch::MtchActor::idNum

PropModel DemoMtch::MtchActor::pMod

double DemoMtch::MtchActor::sCap

vector<double> DemoMtch::MtchActor::vals

VotingRule DemoMtch::MtchActor::vr

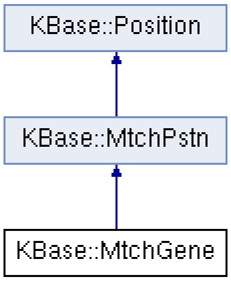
The documentation for this class was generated from the following files:

- **demomtch.h**
- **demomtch.cpp**

KBase::MtchGene Class Reference

`#include <kmodel.h>`

Inheritance diagram for KBase::MtchGene:



Public Member Functions

- **MtchGene** ()
- **~MtchGene** ()
- void **randomize** (**PRNG** *rng)
- **MtchGene** * **mutate** (**PRNG** *rng) const
- tuple< **MtchGene** *, **MtchGene** * > **cross** (const **MtchGene** *g2, **PRNG** *rng) const
- void **show** () const
- bool **equiv** (const **MtchGene** *g2) const
- void **setState** (vector< **Actor** * > as, vector< **MtchPstn** * > ps)

Protected Member Functions

- void **copySelf** (**MtchGene** *) const

Protected Attributes

- vector< **Actor** * > **actrs**
- vector< **MtchPstn** * > **pstns**

Additional Inherited Members

Constructor & Destructor Documentation

KBase::MtchGene::MtchGene ()

KBase::MtchGene::~MtchGene ()

Member Function Documentation

void KBase::MtchGene::copySelf (MtchGene *   *mg2*) const`[protected]`

tuple< MtchGene *, MtchGene * > KBase::MtchGene::cross (const MtchGene *   *g2*, PRNG *   *rng*) const

bool KBase::MtchGene::equiv (const MtchGene *   *g2*) const

MtchGene * KBase::MtchGene::mutate (PRNG *   *rng*) const

void KBase::MtchGene::randomize (PRNG *   *rng*)

void KBase::MtchGene::setState (vector< Actor * >   *as*, vector< MtchPstn * >   *ps*)

void KBase::MtchGene::show () const

Member Data Documentation

vector<Actor*> KBase::MtchGene::actrs`[protected]`

vector<MtchPstn*> KBase::MtchGene::pstns`[protected]`

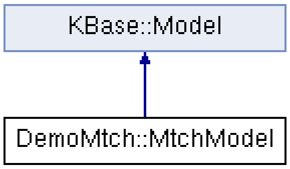
The documentation for this class was generated from the following files:

- **kmodel.h**
- **kposition.cpp**

DemoMtch::MtchModel Class Reference

`#include <demomtch.h>`

Inheritance diagram for DemoMtch::MtchModel:



Public Member Functions

- **MtchModel** (**PRNG** ***rng**)
- virtual **~MtchModel** ()

Static Public Member Functions

- static **MtchModel** * **randomMS** (unsigned int numA, unsigned int numI, VotingRule vr, **MtchActor::PropModel** pMod, **PRNG** ***rng**)

Public Attributes

- unsigned int **numItm**
- unsigned int **numCat**

Additional Inherited Members

Constructor & Destructor Documentation

DemoMtch::MtchModel::MtchModel (PRNG * *rng*)

DemoMtch::MtchModel::~MtchModel () `[virtual]`

Member Function Documentation

MtchModel * DemoMtch::MtchModel::randomMS (unsigned int *numA*, unsigned int *numI*, VotingRule *vr*, MtchActor::PropModel *pMod*, PRNG * *rng*) `[static]`

Member Data Documentation

unsigned int DemoMtch::MtchModel::numCat

unsigned int DemoMtch::MtchModel::numItm

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

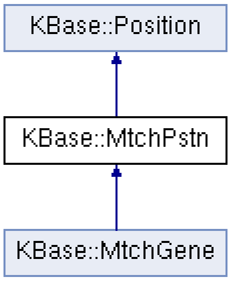
- **demomtch.h**

- **demomtch.cpp**

KBase::MtchPstn Class Reference

`#include <kmodel.h>`

Inheritance diagram for KBase::MtchPstn:



Public Member Functions

- **MtchPstn** ()
- virtual **~MtchPstn** ()
- virtual vector< **MtchPstn** > **neighbors** (unsigned int nVar) const

Public Attributes

- unsigned int **numItm**
- unsigned int **numCat**
- vector< unsigned int > **match**

Constructor & Destructor Documentation

KBase::MtchPstn::MtchPstn ()

KBase::MtchPstn::~MtchPstn () `[virtual]`

Member Function Documentation

vector< MtchPstn > KBase::MtchPstn::neighbors (unsigned int *nVar*) const `[virtual]`

Member Data Documentation

vector<unsigned int> KBase::MtchPstn::match

unsigned int KBase::MtchPstn::numCat

unsigned int KBase::MtchPstn::numItm

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

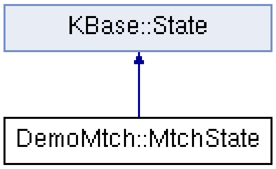
- **kmodel.h**

- **kposition.cpp**

DemoMtch::MtchState Class Reference

```
#include <demomtch.h>
```

Inheritance diagram for DemoMtch::MtchState:



Public Member Functions

- **MtchState** (**Model** *mod)
- **~MtchState** ()
- **KMatrix actrCaps** () const
- **KMatrix pDist** (int persp) const
- void **setAUtil** (ReportingLevel rl)
- **MtchState** * **stepSUSN** ()
- **MtchState** * **stepBCN** ()

Protected Member Functions

- **MtchState** * **doSUSN** (ReportingLevel rl) const
- **MtchState** * **doBCN** (ReportingLevel rl) const

Additional Inherited Members

Constructor & Destructor Documentation

DemoMtch::MtchState::MtchState (Model * *mod*)

DemoMtch::MtchState::~MtchState ()

Member Function Documentation

KMatrix DemoMtch::MtchState::actrCaps () const

MtchState * DemoMtch::MtchState::doBCN (ReportingLevel *rl*) const `[protected]`

MtchState * DemoMtch::MtchState::doSUSN (ReportingLevel *rl*) const `[protected]`

KMatrix DemoMtch::MtchState::pDist (int *persp*) const `[virtual]`

Implements **KBase::State** (*p.58*).

void DemoMtch::MtchState::setAUtil (ReportingLevel *rl*)

MtchState * DemoMtch::MtchState::stepBCN ()

MtchState * DemoMtch::MtchState::stepSUSN ()

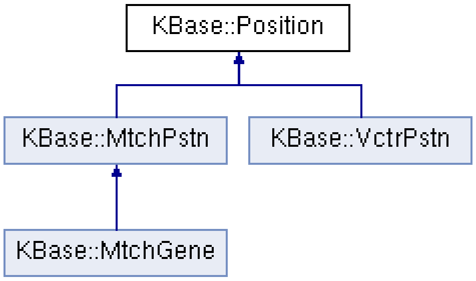
The documentation for this class was generated from the following files:

- **demomtch.h**
- **demomtch.cpp**

KBase::Position Class Reference

`#include <kmodel.h>`

Inheritance diagram for KBase::Position:



Public Member Functions

- **Position** ()
- virtual **~Position** ()

Constructor & Destructor Documentation

KBase::Position::Position ()

KBase::Position::~Position () `[virtual]`

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

- **kmodel.h**
- **kmodel.cpp**

KBase::PRNG Class Reference

`#include <prng.h>`

Public Member Functions

- **PRNG** ()
- virtual **~PRNG** ()
- uint64_t **uniform** ()
- double **uniform** (double a, double b)
- vector< bool > **bits** (unsigned int nb)
- uint64_t **setSeed** (uint64_t)

Protected Attributes

- mt19937_64 **mt**

Constructor & Destructor Documentation

KBase::PRNG::PRNG ()

KBase::PRNG::~PRNG () `[virtual]`

Member Function Documentation

vector< bool > KBase::PRNG::bits (unsigned int *nb*)

uint64_t KBase::PRNG::setSeed (uint64_t *s*)

uint64_t KBase::PRNG::uniform ()

double KBase::PRNG::uniform (double *a*, double *b*)

Member Data Documentation

mt19937_64 KBase::PRNG::mt `[protected]`

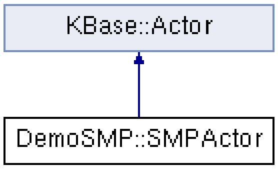
The documentation for this class was generated from the following files:

- **prng.h**
- **prng.cpp**

DemoSMP::SMPActor Class Reference

`#include <demosmp.h>`

Inheritance diagram for DemoSMP::SMPActor:



Public Types

- enum **InterVecBrgn** { **InterVecBrgn::S1P1**, **InterVecBrgn::S2P2**, **InterVecBrgn::S2PMax** }

Public Member Functions

- **SMPActor** (string n, string d)
- **~SMPActor** ()
- double **vote** (unsigned int p1, unsigned int p2, const **State** *st) const
- virtual double **vote** (const **Position** *ap1, const **Position** *ap2, const **SMPState** *as1) const
- double **posUtil** (const **Position** *ap1, const **SMPState** *as1) const
- void **randomize** (**PRNG** *rng, unsigned int numD)

Static Public Member Functions

- static **BargainSMP** * **interpolateBrgn** (const **SMPActor** *ai, const **SMPActor** *aj, const **VctrPstn** *posI, const **VctrPstn** *posJ, double prbI, double prbJ, **InterVecBrgn** ivb)

Public Attributes

- double **sCap**
- **KMatrix vSal**
- VotingRule **vr**

Static Protected Member Functions

- static void **interpBrgnSnPm** (unsigned int n, unsigned int m, double tik, double sik, double prbI, double tjk, double sjk, double prbJ, double &bik, double &bjk)
- static void **interpBrgnS2PMax** (double tik, double sik, double prbI, double tjk, double sjk, double prbJ, double &bik, double &bjk)

Member Enumeration Documentation

enum DemoSMP::SMPActor::InterVecBrgn `[strong]`

Enumerator

S1P1
S2P2
S2PMax

Constructor & Destructor Documentation

DemoSMP::SMPActor::SMPActor (string *n*, string *d*)

DemoSMP::SMPActor::~SMPActor ()

Member Function Documentation

void DemoSMP::SMPActor::interpBrgnS2PMax (double *tik*, double *sik*, double *prbI*, double *tjk*, double *sjk*, double *prbJ*, double & *bik*, double & *bjk*)`[static], [protected]`

void DemoSMP::SMPActor::interpBrgnSnPm (unsigned int *n*, unsigned int *m*, double *tik*, double *sik*, double *prbI*, double *tjk*, double *sjk*, double *prbJ*, double & *bik*, double & *bjk*)`[static], [protected]`

BargainSMP * DemoSMP::SMPActor::interpolateBrgn (const SMPActor * *ai*, const SMPActor * *aj*, const VctrPstn * *posI*, const VctrPstn * *posJ*, double *prbI*, double *prbJ*, InterVecBrgn *ivb*)`[static]`

double DemoSMP::SMPActor::posUtil (const Position * *ap1*, const SMPState * *as1*) const

void DemoSMP::SMPActor::randomize (PRNG * *rng*, unsigned int *numD*)

double DemoSMP::SMPActor::vote (unsigned int *p1*, unsigned int *p2*, const State * *st*) const`[virtual]`

Implements **KBase::Actor** (*p.18*).

double DemoSMP::SMPActor::vote (const Position * *ap1*, const Position * *ap2*, const SMPState * *as1*) const`[virtual]`

Member Data Documentation

double DemoSMP::SMPActor::sCap

VotingRule DemoSMP::SMPActor::vr

KMatrix DemoSMP::SMPActor::vSal

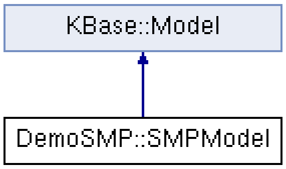
The documentation for this class was generated from the following files:

- **demosmp.h**
- **demosmp.cpp**

DemoSMP::SMPModel Class Reference

`#include <demosmp.h>`

Inheritance diagram for DemoSMP::SMPModel:



Public Member Functions

- **SMPModel** (**PRNG** ***rng**)
- virtual **~SMPModel** ()
- void **showVPHistory** () const

Static Public Member Functions

- static double **bsUtil** (double d, double R)
- static double **bvDiff** (const **KMatrix** &d, const **KMatrix** &s)
- static double **bvUtil** (const **KMatrix** &d, const **KMatrix** &s, double R)
- static **SMPModel** * **readCSV** (string fName, **PRNG** ***rng**)
- static **SMPModel** * **initModel** (vector< string > aName, vector< string > aDesc, vector< string > dName, **KMatrix** cap, **KMatrix** pos, **KMatrix** sal, **PRNG** ***rng**)
- static double **stateDist** (const **SMPState** *s1, const **SMPState** *s2)

Public Attributes

- unsigned int **numDim**
- vector< string > **dimName**

Protected Member Functions

- void **addDim** (string dn)

Additional Inherited Members

Constructor & Destructor Documentation

DemoSMP::SMPModel::SMPModel (PRNG * *rng*)

DemoSMP::SMPModel::~SMPModel () `[virtual]`

Member Function Documentation

void DemoSMP::SMPModel::addDim (string *dn*)`[protected]`

double DemoSMP::SMPModel::bsUtil (double *d*, double *R*)`[static]`

double DemoSMP::SMPModel::bvDiff (const KMatrix & *d*, const KMatrix & *s*)`[static]`

double DemoSMP::SMPModel::bvUtil (const KMatrix & *d*, const KMatrix & *s*, double *R*)`[static]`

SMPModel * DemoSMP::SMPModel::initModel (vector< string > *aName*, vector< string > *aDesc*, vector< string > *dName*, KMatrix *cap*, KMatrix *pos*, KMatrix *sal*, PRNG * *rng*)`[static]`

static SMPModel* DemoSMP::SMPModel::readCSV (string *fName*, PRNG * *rng*)`[static]`

void DemoSMP::SMPModel::showVPHistory () const

double DemoSMP::SMPModel::stateDist (const SMPState * *s1*, const SMPState * *s2*)`[static]`

Member Data Documentation

vector<string> DemoSMP::SMPModel::dimName

unsigned int DemoSMP::SMPModel::numDim

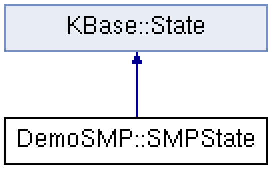
The documentation for this class was generated from the following files:

- **demosmp.h**
- **demosmp.cpp**

DemoSMP::SMPState Class Reference

`#include <demosmp.h>`

Inheritance diagram for DemoSMP::SMPState:



Public Types

- enum **BigRRange** { **BigRRange::Min**, **BigRRange::Mid**, **BigRRange::Max** }
- enum **BigRAdjust** { **BigRAdjust::None**, **BigRAdjust::Half**, **BigRAdjust::Full** }

Public Member Functions

- **SMPState** (**Model** *m)
- virtual **~SMPState** ()
- virtual void **setDiff** ()
- virtual void **setAUtil** (ReportingLevel rl)
- double **estNRA** (unsigned int h, unsigned int i, **SMPState::BigRAdjust** ra) const
- **KMatrix actrCaps** () const
- **SMPState** * **stepBCN** ()
- virtual void **addPstn** (**Position** *p)
- virtual **KMatrix pDist** (int persp) const
- void **showBargains** (const vector< vector< **BargainSMP** * > > &brgns) const

Public Attributes

- **KMatrix nra**
- function< shared_ptr< void >const **Actor** *ai, const **State** *s)> **bestTarget**
- function< shared_ptr< void >const **Actor** *aInit, const **Actor** *aRcvr, shared_ptr< void > btData, const **State** *s)> **bargain**

Protected Member Functions

- **SMPState** * **doBCN** () const
- tuple< double, double > **probEduChlg** (unsigned int h, unsigned int k, unsigned int i, unsigned int j) const
- tuple< int, double, double > **bestChallenge** (unsigned int i) const

Static Protected Member Functions

- static **KMatrix bigRfromProb** (const **KMatrix** &p, **BigRRange** rr)

Protected Attributes

- **KMatrix diff**

Member Enumeration Documentation

enum DemoSMP::SMPState::BigRAdjust `[strong]`

Enumerator

None
Half
Full

enum DemoSMP::SMPState::BigRRange`[strong]`

Enumerator

Min
Mid
Max

Constructor & Destructor Documentation

DemoSMP::SMPState::SMPState (Model * *m*)

DemoSMP::SMPState::~SMPState ()`[virtual]`

Member Function Documentation

KMatrix DemoSMP::SMPState::actrCaps () const

void DemoSMP::SMPState::addPstn (Position * *p*)`[virtual]`

Reimplemented from **KBase::State** (*p.58*).

tuple< int, double, double > DemoSMP::SMPState::bestChallenge (unsigned int *i*) const`[protected]`

KMatrix DemoSMP::SMPState::bigRfromProb (const KMatrix & *p*, BigRRange *rr*)`[static]`, `[protected]`

SMPState * DemoSMP::SMPState::doBCN () const`[protected]`

double DemoSMP::SMPState::estNRA (unsigned int *h*, unsigned int *i*, SMPState::BigRAdjust *ra*) const

KMatrix DemoSMP::SMPState::pDist (int *persp*) const`[virtual]`

Implements **KBase::State** (*p.58*).

tuple< double, double > DemoSMP::SMPState::probEduChlg (unsigned int ***h*****, unsigned int** ***k*****, unsigned int** ***i*****, unsigned int** ***j*****) const**`[protected]`

void DemoSMP::SMPState::setAUtil (ReportingLevel ***rl*****)**`[virtual]`

void DemoSMP::SMPState::setDiff ()`[virtual]`

void DemoSMP::SMPState::showBargains (const vector< vector< BargainSMP * > > & ***brgns*****) const**

SMPState * DemoSMP::SMPState::stepBCN ()

Member Data Documentation

function<shared_ptr<void>const Actor* aInit, const Actor* aRcvr, shared_ptr<void> btData, const State* s)> DemoSMP::SMPState::bargain

function<shared_ptr<void>const Actor* ai, const State* s)> DemoSMP::SMPState::bestTarget

KMatrix DemoSMP::SMPState::diff`[protected]`

KMatrix DemoSMP::SMPState::nra

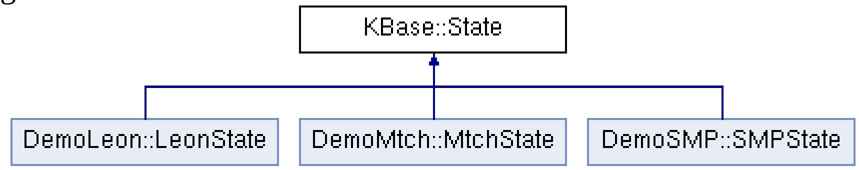
The documentation for this class was generated from the following files:

- **demosmp.h**
- **demosmp.cpp**

KBase::State Class Reference

`#include <kmodel.h>`

Inheritance diagram for KBase::State:



Public Member Functions

- **State** (**Model** *mod)
- virtual **~State** ()
- void **randomizeUtils** (double minU, double maxU, double uNoise)
- void **clear** ()
- virtual void **addPstn** (**Position** *p)
- virtual **KMatrix** **pDist** (int persp) const =0

Public Attributes

- **Model** * **model**
- function< **State** *() > **step**
- vector< **KMatrix** > **aUtil**
- vector< **Position** * > **pstns**

Constructor & Destructor Documentation

KBase::State::State (Model * *mod*)

KBase::State::~State () `[virtual]`

Member Function Documentation

void KBase::State::addPstn (Position * *p*) `[virtual]`

Reimplemented in **DemoSMP::SMPState** (*p.56*).

void KBase::State::clear ()

virtual KMatrix KBase::State::pDist (int *persp*) const `[pure virtual]`

Implemented in **DemoSMP::SMPState** (*p.56*), **DemoLeon::LeonState** (*p.33*), and **DemoMtch::MtchState** (*p.47*).

void KBase::State::randomizeUtils (double *minU*, double *maxU*, double *uNoise*)

Member Data Documentation

vector<KMatrix> KBase::State::aUtil

Model* KBase::State::model

vector<Position*> KBase::State::pstns

function<State* () > KBase::State::step

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

- **kmodel.h**
- **kstate.cpp**

UDemo::TargetedBV Class Reference

```
#include <demo.h>
```

Public Member Functions

- **TargetedBV** ()
- virtual **~TargetedBV** ()
- virtual void **randomize** (**PRNG** *rng)
- virtual **TargetedBV * mutate** (**PRNG** *rng) const
- virtual tuple< **TargetedBV** *, **TargetedBV** * > **cross** (const **TargetedBV** *g2, **PRNG** *rng) const
- virtual void **show** () const
- virtual bool **equiv** (const **TargetedBV** *g2) const
- double **evaluate** ()
- double **tblEval** (double minD, vector< double > weights, vector< **BVec** > tbl) const
- unsigned int **hDist** (**BVec** bv) const

Static Public Member Functions

- static void **setTarget** (**BVec** trgt)
- static **BVec getTarget** ()
- static void **showBits** (**BVec** bv)
- static **BVec randomBV** (**PRNG** *rng, unsigned int nb)

Public Attributes

- **BVec bits**

Static Public Attributes

- static **BVec target**

Constructor & Destructor Documentation

UDemo::TargetedBV::TargetedBV ()

UDemo::TargetedBV::~TargetedBV () `[virtual]`

Member Function Documentation

tuple< TargetedBV *, TargetedBV * > UDemo::TargetedBV::cross (const TargetedBV * *g2*, PRNG * *rng*) const`[virtual]`

bool UDemo::TargetedBV::equiv (const TargetedBV * *g2*) const`[virtual]`

double UDemo::TargetedBV::evaluate ()

vector< bool > UDemo::TargetedBV::getTarget ()`[static]`

unsigned int UDemo::TargetedBV::hDist (BVec *bv*) const

TargetedBV * UDemo::TargetedBV::mutate (PRNG * *rng*) const`[virtual]`

vector< bool > UDemo::TargetedBV::randomBV (PRNG * *rng*, unsigned int *nb*)`[static]`

void UDemo::TargetedBV::randomize (PRNG * *rng*)`[virtual]`

void UDemo::TargetedBV::setTarget (BVec *trgt*)`[static]`

void UDemo::TargetedBV::show () const`[virtual]`

void UDemo::TargetedBV::showBits (BVec *bv*)`[static]`

double UDemo::TargetedBV::tblEval (double *minD*, vector< double > *weights*, vector< BVec > *tbl*) const

Member Data Documentation

BVec UDemo::TargetedBV::bits

vector< bool > UDemo::TargetedBV::target`[static]`

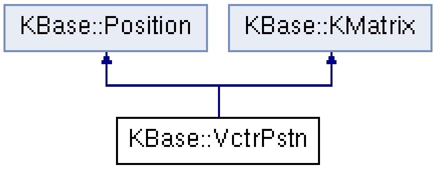
The documentation for this class was generated from the following files:

- **kutils/src/demo.h**
- **kutils/src/demo.cpp**

KBase::VctrPstn Class Reference

`#include <kmodel.h>`

Inheritance diagram for KBase::VctrPstn:



Public Member Functions

- **VctrPstn** ()
- **VctrPstn** (unsigned int nr, unsigned int nc)
- **VctrPstn** (const **KMatrix** &m)
- virtual **~VctrPstn** ()

Additional Inherited Members

Constructor & Destructor Documentation

KBase::VctrPstn::VctrPstn ()

KBase::VctrPstn::VctrPstn (unsigned int *nr*, unsigned int *nc*)

KBase::VctrPstn::VctrPstn (const KMatrix & *m*)

KBase::VctrPstn::~VctrPstn () `[virtual]`

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

- **kmodel.h**
- **kposition.cpp**

KBase::VHCSearch Class Reference

```
#include <hcsearch.h>
```

Public Member Functions

- **VHCSearch** ()
- virtual **~VHCSearch** ()
- tuple< double, **KMatrix**, unsigned int, unsigned int > **run** (**KMatrix** p0, unsigned int iMax, unsigned int sMax, double sTol, double s0, double shrink, double grow, double minStep, **ReportingLevel** rl)

Static Public Member Functions

- static vector< **KMatrix** > **vn1** (const **KMatrix** &m0, double s)
- static vector< **KMatrix** > **vn2** (const **KMatrix** &m0, double s)

Public Attributes

- function< double(const **KMatrix** &)> **eval**
- function< vector< **KMatrix** >const **KMatrix** &, double)> **nghbrs**
- function< void(const **KMatrix** &)> **report**

Constructor & Destructor Documentation

KBase::VHCSearch::VHCSearch ()

KBase::VHCSearch::~VHCSearch () `[virtual]`

Member Function Documentation

tuple< double, KMatrix, unsigned int, unsigned int > KBase::VHCSearch::run (KMatrix *p0*, unsigned int *iMax*, unsigned int *sMax*, double *sTol*, double *s0*, double *shrink*, double *grow*, double *minStep*, ReportingLevel *rl*)

vector< KMatrix > KBase::VHCSearch::vn1 (const KMatrix & *m0*, double *s*) `[static]`

vector< KMatrix > KBase::VHCSearch::vn2 (const KMatrix & *m0*, double *s*) `[static]`

Member Data Documentation

function<double(const KMatrix &)> KBase::VHCSearch::eval

function< vector<KMatrix>const KMatrix &, double)> KBase::VHCSearch::nghbrs

function<void (const KMatrix &)> KBase::VHCSearch::report

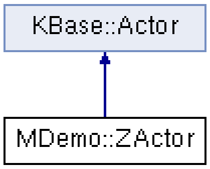
The documentation for this class was generated from the following files:

- **hcsearch.h**
- **hcsearch.cpp**

MDemo::ZActor Class Reference

`#include <zactor.h>`

Inheritance diagram for MDemo::ZActor:



Public Member Functions

- **ZActor** (string n, string d)
- **~ZActor** ()
- double **vote** (unsigned int p1, unsigned int p2, const **State** *st) const
- virtual double **vote** (const **Position** *ap1, const **Position** *ap2) const
- double **posUtil** (const **Position** *ap1) const

Additional Inherited Members

Constructor & Destructor Documentation

MDemo::ZActor::ZActor (string *n*, string *d*)

MDemo::ZActor::~ZActor ()

Member Function Documentation

double MDemo::ZActor::posUtil (const Position * *ap1*) const

double MDemo::ZActor::vote (unsigned int *p1*, unsigned int *p2*, const State * *st*) const`[virtual]`

Implements **KBase::Actor** (*p.18*).

double MDemo::ZActor::vote (const Position * *ap1*, const Position * *ap2*) const`[virtual]`

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

- **zactor.h**
- **zactor.cpp**

File Documentation

demo.cpp File Reference

```
#include "kutils.h"
#include "kmodel.h"
#include "gaopt.h"
#include "hcsearch.h"
#include "demo.h"
#include "demomtch.h"
#include "demoleon.h"
#include "demosmp.h"
```

Namespaces

- **MDemo**

Functions

- void **MDemo::demoPCE** (uint64_t s, **PRNG** *rng)
- void **MDemo::demoSpVSR** (uint64_t s, **PRNG** *rng)
- int **main** (int ac, char **av)

Function Documentation

int main (int *ac*, char ** *av*)

demo.cpp File Reference

```
#include "demo.h"
```

Namespaces

- **UDemo**

Functions

- void **UDemo::show** (string str, const **KMatrix** &m, string fs)
- double **UDemo::nProd** (double x, double y)
- double **UDemo::bsu** (double d, double R)
- double **UDemo::bvu** (const **KBase::KMatrix** &d, const **KBase::KMatrix** &s, double R)
- void **UDemo::demoMatrix** (**PRNG** *rng)
- void **UDemo::demoABG00** (**PRNG** *rng)
- double **UDemo::eNorm** (const **KMatrix** &a, const **KMatrix** &x)
- **KMatrix UDemo::eUnitize** (const **KMatrix** &a, const **KMatrix** &x)
- **KMatrix UDemo::projEllipse** (const **KMatrix** &a, const **KMatrix** &w)
- void **UDemo::demoEllipseLVI** (**PRNG** *rng, unsigned int n)
- tuple< **KMatrix**, **KMatrix**, **KMatrix**, **KMatrix** > **UDemo::antiLemke** (unsigned int n)
- void **UDemo::demoAntiLemke** (**PRNG** *rng, unsigned int n)
- void **UDemo::demoEllipse** (**PRNG** *rng)
- void **UDemo::demoGA** (**PRNG** *rng)
- void **UDemo::demoGHC** (**PRNG** *rng)
- void **UDemo::demoVHC00** (**PRNG** *rng)
- void **UDemo::demoVHC01** (**PRNG** *rng)
- void **UDemo::demoVHC02** (**PRNG** *rng)
- void **UDemo::demoVHC03** (**PRNG** *rng)
- void **UDemo::parallelMatrixMult** (**PRNG** *rng)
- int **main** (int ac, char **av)

Function Documentation

int main (int *ac*, char ** *av*)

demo.h File Reference

```
#include <assert.h>
#include <chrono>
#include <cstring>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include "kutils.h"
#include "prng.h"
#include "kmatrix.h"
#include "kmodel.h"
```

Namespaces

- **MDemo**

demo.h File Reference

```
#include "kutils.h"
#include "prng.h"
#include "kmatrix.h"
#include "gaopt.h"
#include "hcsearch.h"
#include "vimcp.h"
```

Classes

- class **UDemo::TargetedBV**

Namespaces

- **UDemo**

Typedefs

- typedef vector< bool > **UDemo::BVec**

Functions

- double **UDemo::eNorm** (const **KMatrix** &a, const **KMatrix** &x)
- **KMatrix UDemo::eUnitize** (const **KMatrix** &a, const **KMatrix** &x)
- **KMatrix UDemo::projEllipse** (const **KMatrix** &a, const **KMatrix** &w)
- void **UDemo::demoEllipseLVI** (**PRNG** *rng, unsigned int n)
- void **UDemo::demoAntiLemke** (**PRNG** *rng, unsigned int n)

demoleon.cpp File Reference

```
#include "demoleon.h"
```

Namespaces

- **DemoLeon**

Functions

- LeonModel * **DemoLeon::demoSetup** (unsigned int numFctr, unsigned int numCGrp, unsigned int numSect, uint64_t s, **PRNG** *rng)
- void **DemoLeon::demoEUEcon** (uint64_t s, unsigned int numF, unsigned int numG, unsigned int numS, **PRNG** *rng)
- void **DemoLeon::demoMaxEcon** (uint64_t s, unsigned int numF, unsigned int numG, unsigned int numS, **PRNG** *rng)
- int **main** (int ac, char **av)

Variables

- const double **DemoLeon::TolIFD** = 1E-6

Function Documentation

int main (int *ac*, char ** *av*)

demoleon.h File Reference

```
#include <assert.h>
#include <chrono>
#include <cstring>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <tuple>
#include <vector>
#include "kutils.h"
#include "prng.h"
#include "kmatrix.h"
#include "gaopt.h"
#include "hcsearch.h"
#include "kmodel.h"
```

Classes

- class **DemoLeon::LeonActor**
- class **DemoLeon::LeonState**
- class **DemoLeon::LeonModel**

Namespaces

- **DemoLeon**

Functions

- LeonModel * **DemoLeon::demoSetup** (unsigned int numFctr, unsigned int numCGrp, unsigned int numSect, uint64_t s, **PRNG** *rng)
- void **DemoLeon::demoEUEcon** (uint64_t s, PRNG *rng)
- void **DemoLeon::demoMaxEcon** (uint64_t s, PRNG *rng)

demomin.cpp File Reference

```
#include "demomin.h"
```

Namespaces

- **DemoWaterMin**

Functions

- double **DemoWaterMin::waterMinProb** (ReportingLevel rl, const KMatrix &p0)
- void **DemoWaterMin::minProbErr** ()
- int **main** (int ac, char **av)

Function Documentation

int main (int *ac*, char ** *av*)

demomin.h File Reference

```
#include <assert.h>
#include <chrono>
#include <cstring>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <tuple>
#include <vector>
#include "kutils.h"
#include "prng.h"
#include "kmatrix.h"
#include "gaopt.h"
#include "hcsearch.h"
#include "kmodel.h"
```

Namespaces

- **DemoWaterMin**

demomtch.cpp File Reference

```
#include "demomtch.h"
```

Namespaces

- **DemoMtch**

Functions

- bool **DemoMtch::equivMtchPstn** (const MtchPstn &mp1, const MtchPstn &mp2)
- void **DemoMtch::showMtchPstn** (const MtchPstn &mp)
- bool **DemoMtch::stableMtchState** (unsigned int iter, const State *s1)
- void **DemoMtch::demoDivideSweets** (uint64_t s, **PRNG** *rng)
- void **DemoMtch::demoMaxSupport** (uint64_t s, **PRNG** *rng)
- void **DemoMtch::demoMtchSUSN** (uint64_t s, **PRNG** *rng)
- void **DemoMtch::multiMtchSUSN** (uint64_t s, **PRNG** *rng)
- bool **DemoMtch::oneMtchSUSN** (uint64_t s, **PRNG** *rng)
- int **main** (int ac, char **av)

Function Documentation

int main (int *ac*, char ** *av*)

demomtch.h File Reference

```
#include <assert.h>
#include <chrono>
#include <cstring>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <tuple>
#include <vector>
#include "kutils.h"
#include "prng.h"
#include "kmatrix.h"
#include "gaopt.h"
#include "hcsearch.h"
#include "kmodel.h"
```

Classes

- class **DemoMtch::MtchActor**
- class **DemoMtch::MtchState**
- class **DemoMtch::MtchModel**

Namespaces

- **DemoMtch**

Functions

- void **DemoMtch::demoDivideSweets** (uint64_t s, **PRNG** *rng)
- void **DemoMtch::demoMaxSupport** (uint64_t s, **PRNG** *rng)
- void **DemoMtch::demoMtchSUSN** (uint64_t s, **PRNG** *rng)
- void **DemoMtch::multiMtchSUSN** (uint64_t s, **PRNG** *rng)
- bool **DemoMtch::oneMtchSUSN** (uint64_t s, **PRNG** *rng)
- void **DemoMtch::showMtchPstn** (const MtchPstn &mp)
- bool **DemoMtch::stableMtchState** (unsigned int iter, const State *s1)

demosmp.cpp File Reference

```
#include "kutils.h"
#include "kmodel.h"
#include "demo.h"
#include "demosmp.h"
#include "csv_parser.hpp"
```

Namespaces

- **DemoSMP**

Functions

- SMPModel * **DemoSMP::readCSV** (string fName, **PRNG** *rng)
- void **DemoSMP::demoActorUtils** (uint64_t s, **PRNG** *rng)
- void **DemoSMP::demoEUSpatial** (unsigned int numA, unsigned int sDim, uint64_t s, **PRNG** *rng)
- void **DemoSMP::readEUSpatial** (uint64_t seed, string inputCSV, **PRNG** *rng)
- int **main** (int ac, char **av)

Function Documentation

int main (int *ac*, char ** *av*)

demosmp.h File Reference

```
#include "kutils.h"
#include "prng.h"
#include "kmatrix.h"
#include "gaopt.h"
#include "kmodel.h"
```

Classes

- struct **DemoSMP::BargainSMP**
- class **DemoSMP::SMPActor**
- class **DemoSMP::SMPState**
- class **DemoSMP::SMPModel**

Namespaces

- **DemoSMP**

Functions

- void **DemoSMP::demoActorUtils** (uint64_t s, **PRNG** *rng)
- void **DemoSMP::demoEUSpatial** (unsigned int numA, unsigned int sDim, uint64_t s, **PRNG** *rng)

Variables

- const string **DemoSMP::appVersion** = "0.1"

gaopt.cpp File Reference

```
#include <assert.h>
#include <iostream>
#include <tuple>
#include "kutils.h"
#include "prng.h"
#include "gaopt.h"
```

Namespaces

- **KBase**

Functions

- unsigned int **KBase::crossSite** (PRNG *rng, unsigned int nc)

gaopt.h File Reference

```
#include <assert.h>
#include <chrono>
#include <functional>
#include <iostream>
#include <string>
#include <tuple>
#include <vector>
#include "prng.h"
#include "kutils.h"
```

Classes

- class **KBase::GAOpt< GAP >**

Namespaces

- **KBase**

Functions

- unsigned int **KBase::crossSite** (PRNG *rng, unsigned int nc)

hcsearch.cpp File Reference

```
#include "kutils.h"
#include "hcsearch.h"
```

Namespaces

- **KBase**

hcsearch.h File Reference

```
#include <functional>
#include <iostream>
#include <tuple>
#include <vector>
#include "kutils.h"
#include "kmatrix.h"
```

Classes

- class **KBase::VHCSearch**
- class **KBase::GHCSearch< HCP >**

Namespaces

- **KBase**

kmatrix.cpp File Reference

```
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <iostream>
#include <string.h>
#include <vector>
#include "prng.h"
#include "kmatrix.h"
```

Namespaces

- **KBase**

Functions

- KMatrix **KBase::trans** (const KMatrix &m1)
- double **KBase::norm** (const KMatrix &m1)
- double **KBase::sum** (const KMatrix &m1)
- double **KBase::mean** (const KMatrix &m1)
- double **KBase::stdv** (const KMatrix &m1)
- double **KBase::maxAbs** (const KMatrix &m)
- tuple< unsigned int, unsigned int > **KBase::ndxMaxAbs** (const KMatrix &m)
- double **KBase::lCorr** (const KMatrix &m1, const KMatrix &m2)
- double **KBase::dot** (const KMatrix &m1, const KMatrix &m2)
- KMatrix **KBase::operator+** (const KMatrix &m1, double x)
- KMatrix **KBase::operator-** (const KMatrix &m1, double x)
- bool **KBase::sameShape** (const KMatrix &m1, const KMatrix &m2)
- KMatrix **KBase::operator+** (const KMatrix &m1, const KMatrix &m2)
- KMatrix **KBase::operator-** (const KMatrix &m1, const KMatrix &m2)
- KMatrix **KBase::operator*** (double x, const KMatrix &m1)
- KMatrix **KBase::operator/** (const KMatrix &m1, double x)
- KMatrix **KBase::operator*** (const KMatrix &m1, const KMatrix &m2)
- KMatrix **KBase::inv** (const KMatrix &m)
- KMatrix **KBase::iMat** (unsigned int n)
- KMatrix **KBase::makePerp** (const KMatrix &x, const KMatrix &p)
- KMatrix **KBase::joinH** (const KMatrix &mL, const KMatrix &mR)
- KMatrix **KBase::joinV** (const KMatrix &mT, const KMatrix &mB)

kmatrix.h File Reference

```
#include <cstdint>
#include <functional>
#include <tuple>
#include <vector>
#include "kutils.h"
```

Classes

- class **KBase::KMatrix**

Namespaces

- **KBase**

Functions

- KMatrix **KBase::trans** (const KMatrix &m1)
- double **KBase::norm** (const KMatrix &m1)
- double **KBase::sum** (const KMatrix &m1)
- double **KBase::mean** (const KMatrix &m1)
- double **KBase::stdv** (const KMatrix &m1)
- double **KBase::maxAbs** (const KMatrix &m)
- tuple< unsigned int, unsigned int > **KBase::ndxMaxAbs** (const KMatrix &m)
- double **KBase::dot** (const KMatrix &m1, const KMatrix &m2)
- double **KBase::lCorr** (const KMatrix &m1, const KMatrix &m2)
- KMatrix **KBase::inv** (const KMatrix &m)
- KMatrix **KBase::iMat** (unsigned int n)
- KMatrix **KBase::makePerp** (const KMatrix &x, const KMatrix &p)
- KMatrix **KBase::joinH** (const KMatrix &mL, const KMatrix &mR)
- KMatrix **KBase::joinV** (const KMatrix &mT, const KMatrix &mB)
- KMatrix **KBase::operator+** (const KMatrix &m1, const KMatrix &m2)
- KMatrix **KBase::operator+** (const KMatrix &m1, double x)
- KMatrix **KBase::operator-** (const KMatrix &m1, const KMatrix &m2)
- KMatrix **KBase::operator-** (const KMatrix &m1, double x)
- KMatrix **KBase::operator*** (double x, const KMatrix &m1)
- KMatrix **KBase::operator/** (const KMatrix &m1, double x)
- bool **KBase::sameShape** (const KMatrix &m1, const KMatrix &m2)
- KMatrix **KBase::operator*** (const KMatrix &m1, const KMatrix &m2)

kmodel.cpp File Reference

```
#include <assert.h>
#include <iostream>
#include "kmodel.h"
```

Namespaces

- **KBase**

Functions

- string **KBase::vrName** (VotingRule vr)
- string **KBase::tpcName** (ThirdPartyCommit tpc)

kmodel.h File Reference

```
#include "kutils.h"
#include "kmatrix.h"
#include "prng.h"
```

Classes

- class **KBase::Model**
- class **KBase::State**
- class **KBase::Actor**
- class **KBase::Position**
- class **KBase::VctrPstn**
- class **KBase::MtchPstn**
- class **KBase::MtchGene**

Namespaces

- **KBase**

Enumerations

- enum **KBase::VotingRule** : char { **KBase::VotingRule::Binary**, **KBase::VotingRule::PropBin**, **KBase::VotingRule::Proportional**, **KBase::VotingRule::PropCbc**, **KBase::VotingRule::Cubic** }
- enum **KBase::ThirdPartyCommit** { **KBase::ThirdPartyCommit::None**, **KBase::ThirdPartyCommit::Semi**, **KBase::ThirdPartyCommit::Full** }

Functions

- string **KBase::vrName** (VotingRule vr)
- string **KBase::tpcName** (ThirdPartyCommit tpc)

kposition.cpp File Reference

```
#include <iostream>
#include "gaopt.h"
#include "kmodel.h"
```

Namespaces

- **KBase**

kstate.cpp File Reference

```
#include "kmodel.h"
```

Namespaces

- **KBase**

kutils.cpp File Reference

```
#include <assert.h>
#include <iostream>
#include <tuple>
#include "kutils.h"
#include "prng.h"
```

Namespaces

- **KBase**

Functions

- std::chrono::time_point< std::chrono::system_clock, std::chrono::system_clock::duration > **KBase::displayProgramStart** ()
- void **KBase::displayProgramEnd** (std::chrono::time_point< std::chrono::system_clock, std::chrono::system_clock::duration > &sTime)
- double **KBase::rescale** (double x, double x0, double x1, double y0, double y1)

kutils.h File Reference

```
#include <assert.h>
#include <chrono>
#include <cstdint>
#include <cstdlib>
#include <cstring>
#include <iostream>
#include <functional>
#include <future>
#include <math.h>
#include <memory>
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <thread>
#include <tuple>
#include <vector>
```

Classes

- class **KBase::KException**

Namespaces

- **KBase**

Enumerations

- enum **KBase::ReportingLevel** { **KBase::ReportingLevel::Silent**, **KBase::ReportingLevel::Low**, **KBase::ReportingLevel::Medium**, **KBase::ReportingLevel::High**, **KBase::ReportingLevel::Debugging** }

Functions

- std::chrono::time_point< std::chrono::system_clock, std::chrono::system_clock::duration > **KBase::displayProgramStart** ()
- void **KBase::displayProgramEnd** (std::chrono::time_point< std::chrono::system_clock, std::chrono::system_clock::duration > &sTime)
- double **KBase::rescale** (double x, double x0, double x1, double y0, double y1)
- template<typename T > T **KBase::popBack** (vector< T > &v)

prng.cpp File Reference

```
#include <assert.h>
#include "prng.h"
```

Namespaces

- **KBase**

Functions

- uint64_t **KBase::qTrans** (uint64_t s)

prng.h File Reference

```
#include <cstdint>
#include <random>
#include "kutils.h"
```

Classes

- class **KBase::PRNG**

Namespaces

- **KBase**

Functions

- uint64_t **KBase::qTrans** (uint64_t s)

README.md File Reference

vimcp.cpp File Reference

```
#include "vimcp.h"
```

Namespaces

- **KBase**

Functions

- KMatrix **KBase::projPos** (const KMatrix &w)
- KMatrix **KBase::projBox** (const KMatrix &lb, const KMatrix &ub, const KMatrix &w)
- tuple< KMatrix, unsigned int, KMatrix > **KBase::viABG** (const KMatrix &xInit, function< KMatrix(const KMatrix &x)> F, function< KMatrix(const KMatrix &x)> P, double beta, double thresh, unsigned int iMax)
- tuple< KMatrix, unsigned int, KMatrix > **KBase::solveLVI_BSHe96** (const KMatrix &M, const KMatrix &q, function< KMatrix(const KMatrix &)> pK, KMatrix u0, const double eps, const unsigned int iMax)

vimcp.h File Reference

```
#include <assert.h>
#include <functional>
#include <iostream>
#include <tuple>
#include "kutils.h"
#include "kmatrix.h"
#include "prng.h"
```

Namespaces

- **KBase**

Functions

- tuple< **KMatrix**, **KMatrix**, **KMatrix**, **KMatrix** > **KBase::antiLemke** (unsigned int n)
- KMatrix **KBase::projPos** (const KMatrix &w)
- KMatrix **KBase::projBox** (const KMatrix &lb, const KMatrix &ub, const KMatrix &w)
- tuple< KMatrix, unsigned int, KMatrix > **KBase::viABG** (const KMatrix &xInit, function< KMatrix(const KMatrix &x)> F, function< KMatrix(const KMatrix &x)> P, double beta, double thresh, unsigned int iMax)
- tuple< KMatrix, unsigned int, KMatrix > **KBase::solveLVI_BSHe96** (const KMatrix &M, const KMatrix &q, function< KMatrix(const KMatrix &)> pK, KMatrix u0, const double eps, const unsigned int iMax)

zactor.cpp File Reference

```
#include "kutils.h"
#include "kmodel.h"
#include "demo.h"
#include "zactor.h"
```

Namespaces

- **MDemo**

zactor.h File Reference

```
#include <assert.h>
#include <chrono>
#include <cstring>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <tuple>
#include <vector>
#include "kutils.h"
#include "prng.h"
#include "kmatrix.h"
#include "gaopt.h"
#include "kmodel.h"
```

Classes

- class **MDemo::ZActor**

Namespaces

- **MDemo**

Index