My Project

Generated by Doxygen 1.9.4

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

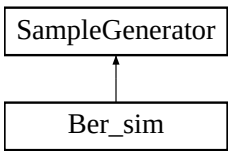
Chapter 4

Class Documentation

4.1 Ber_sim Class Reference

`#include <Ber_sim.h>`

Inheritance diagram for Ber_sim:



Public Member Functions

- **Ber_sim** (int n1=0, double p1=0)

  Set parametrs of the distribution with this constructor.
- **Ber_sim** (Distribution &d)

  Set the distribution itself with this constructor.
- void simulate (int volume) override

  .simulate(volume) writes "volume" realizations of random value of Bin(m,p) in "sample".

Additional Inherited Members

4.1.1 Detailed Description

A class inheriting SampleGenerator and overriding virtual simulating method with Bernoulli method. There is his brother, MOF_sim class that works faster with probabilities far from 0 or 1.

4.1.2 Member Function Documentation

4.1.2.1 simulate()

```
void Ber_sim::simulate (
            int volume )  [inline], [override], [virtual]
```

.simulate(volume) writes "volume" realizations of random value of Bin(m,p) in "sample".

Realizations will be modelled by Bernoulli method.

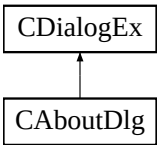
Implements SampleGenerator.

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

- Ber_sim.h

4.2 CAboutDlg Class Reference

Inheritance diagram for CAboutDlg:



Protected Member Functions

- virtual void **DoDataExchange** (CDataExchange ∗pDX)

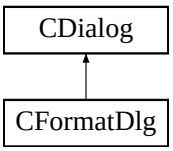
The documentation for this class was generated from the following file:

- Histogram.cpp

4.3 CFormatDlg Class Reference

```
#include <CFormatDlg.h>
```

Inheritance diagram for CFormatDlg:



Public Member Functions

- **CFormatDlg** (CWnd *pParent=nullptr)

Public Attributes

- int **m_n_par**
- double **m_p_par**
- int **m_vol_par**
- int **m_radio**

Protected Member Functions

- virtual void **DoDataExchange** (CDataExchange *pDX)

4.3.1 Detailed Description

A class of a dialog built for collecting parametrs of the binomial distribution in null hypothesis, sample size and method of modelling.

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

- CFormatDlg.h
- CFormatDlg.cpp

4.4 Chi_sq Class Reference

`#include <Chi_sq.h>`

Public Member Functions

- **Chi_sq** ()

  Empty constructor. Fill the data in with the next method.
- void **Set_Data** (int *sample1, int n1, Distribution &d0)

  Method for filling fields that will not be calculated.
- void **Calc_chi_sq** ()

  Method that calculates all the other fields.

Public Attributes

- int **n**

  Sample size that must be given.
- Distribution **d**

  Distribution that must be given. It will provide expected probabilities and frequencies.
- int * **sample**

  Sample that must be given.
- double * **ExpectedPr**
- double * **ExpectedFr**
- int * **hist**

  Empirical frequencies that will be calculated with the sample in basis.
- double **chi_sq**

  The value of statistics of the chi-square criterion that will be calculated.
- int **max_h**

  Maximum of the empirical frequencies that will be calculated.
- double **max_ex_fr**

  Maximum of the expected frequencies that will be calculated.
- double **p_level**
- int **freedom_degree**

4.4.1 Detailed Description

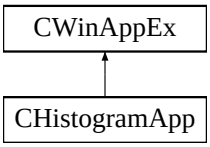
A class made for convenient calculating of statistics of the chi-square criterion including computing of expected probabilities and frequencies, degree of freedom and p-level. All fields except "d", "n" and "sample" will be calculated.

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

- Chi_sq.h
- Chi_sq.cpp

4.5 CHistogramApp Class Reference

Inheritance diagram for CHistogramApp:



Public Member Functions

- virtual BOOL **InitInstance** ()
- virtual void **PreLoadState** ()
- virtual void **LoadCustomState** ()
- virtual void **SaveCustomState** ()
- afx_msg void **OnAppAbout** ()

Public Attributes

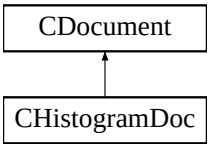
- UINT **m_nAppLook**
- BOOL **m_bHiColorIcons**

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

- Histogram.h
- Histogram.cpp

4.6 CHistogramDoc Class Reference

Inheritance diagram for CHistogramDoc:



Public Member Functions

- virtual BOOL **OnNewDocument** ()
- virtual void **Serialize** (CArchive &ar)
- afx_msg void **OnFormat** ()
- afx_msg void **Onplevels** ()
- afx_msg void **On32775** ()

Public Attributes

- SampleGenerator * **generator**

  It is present for accsess to the "simulate" method.
- Distribution **h0**

  Distribution of the null hypothesis of agreement with the type of distribution.
- Distribution **h1**

  Distribution of the alternative hypothesis.
- Chi_sq **CHI_c**

  It is present for convenient calculating of the chi-square and much more. Read comments in "Chi_sq.h" for more.
- int **form_n**

  It is present to remember sample size that user will enter into dialogues.
- int **form_radio**

  It is present to remember modelling method that user will choose in dialogues.
- int cases
- double **p_lev_alpha**

  It is present to remember significance level that user will enter into dialogues.
- int **p_lev_ss**

  It is present to remember size of sample containing p-levels that user will enter into dialogues.
- double * **p_values**
- double **betas** [12]

  Values of the power that will be calculated and shown.

4.6.1 Member Data Documentation

4.6.1.1 cases

```
int CHistogramDoc::cases
```

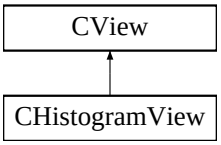
Current state of the program: 1 - working on histogram, 2 - working on p-levels or power distribution, 3 - working on dependence of the power from the sample size.

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

- HistogramDoc.h
- HistogramDoc.cpp

4.7 CHistogramView Class Reference

Inheritance diagram for CHistogramView:



Public Member Functions

- CHistogramDoc ∗ **GetDocument** () const
- virtual void **OnDraw** (CDC ∗pDC)
- virtual BOOL **PreCreateWindow** (CREATESTRUCT &cs)

Protected Member Functions

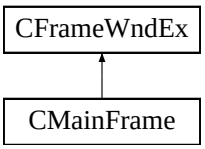
- afx_msg void **OnFilePrintPreview** ()
- afx_msg void **OnRButtonUp** (UINT nFlags, CPoint point)
- afx_msg void **OnContextMenu** (CWnd ∗pWnd, CPoint point)

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

- HistogramView.h
- HistogramView.cpp

4.8 CMainFrame Class Reference

Inheritance diagram for CMainFrame:



Public Member Functions

- virtual BOOL **PreCreateWindow** (CREATESTRUCT &cs)
- virtual BOOL **LoadFrame** (UINT nIDResource, DWORD dwDefaultStyle=WS_OVERLAPPEDWINDOW|FWS↩ _ADDTOTITLE, CWnd ∗pParentWnd=nullptr, CCreateContext ∗pContext=nullptr)

Protected Member Functions

- afx_msg int **OnCreate** (LPCREATESTRUCT lpCreateStruct)
- afx_msg void **OnViewCustomize** ()
- afx_msg LRESULT **OnToolbarCreateNew** (WPARAM wp, LPARAM lp)
- afx_msg void **OnApplicationLook** (UINT id)
- afx_msg void **OnUpdateApplicationLook** (CCmdUI ∗pCmdUI)

Protected Attributes

- CMFCMenuBar **m_wndMenuBar**
- CMFCToolBar **m_wndToolBar**
- CMFCStatusBar **m_wndStatusBar**
- CMFCToolBarImages **m_UserImages**

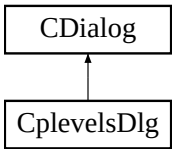
The documentation for this class was generated from the following files:

- MainFrm.h
- MainFrm.cpp

4.9 CplevelsDlg Class Reference

`#include <CplevelsDlg.h>`

Inheritance diagram for CplevelsDlg:



Public Member Functions

- **CplevelsDlg** (CWnd ∗pParent=nullptr)

Public Attributes

- int **m_H0_m**
- double **m_H0_p**
- int **m_H1_m**
- double **m_H1_p**
- int **m_p_val_sam_size**
- double **m_alpha**
- int **m_n**
- int **m_meth**

Protected Member Functions

- virtual void **DoDataExchange** (CDataExchange ∗pDX)

4.9.1 Detailed Description

A class of a dialog built for collecting parameters of distributions in hypotheses, significance level, sample size and method of modelling Bin(m,p) distribution, size of the sample containing p-values.

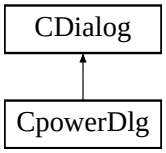
The documentation for this class was generated from the following files:

- CplevelsDlg.h
- CplevelsDlg.cpp

4.10 CpowerDlg Class Reference

`#include <CpowerDlg.h>`

Inheritance diagram for CpowerDlg:



Public Member Functions

- **CpowerDlg** (CWnd ∗pParent=nullptr)

Public Attributes

- int **m_H0_m**
- double **m_H0_p**
- int **m_H1_m**
- double **m_H1_p**
- double **m_alpha**

Protected Member Functions

- virtual void **DoDataExchange** (CDataExchange ∗pDX)

4.10.1 Detailed Description

A class of a dialog built for collecting parameters of distributions in hypotheses and significance level.

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

- CpowerDlg.h
- CpowerDlg.cpp

4.11 Distribution Class Reference

`#include <Disrtibution.h>`

Public Member Functions

- int **get_m** ()
- void **set_m** (int m1)
- double **get_p** ()
- void **set_p** (double p1)
- **Distribution** (int n1=1, double p1=0.1)

  Default constructor will make a Bin(0,0) Distribution object.
- **Distribution** (Distribution ∗d)
- void probabilities ()

Public Attributes

- double ∗ **expected_prob**

4.11.1 Detailed Description

A class containing parametrs of the binomial distribution.

4.11.2 Member Function Documentation

4.11.2.1 probabilities()

```
void Distribution::probabilities ( )  [inline]
```

Method .probabilities() will write a (m+1) theoretical probability of the Bin(m,p) in the "expected_prob".

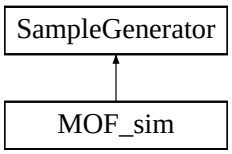
The documentation for this class was generated from the following file:

- Disrtibution.h

4.12 MOF_sim Class Reference

```
#include <MOF_sim.h>
```

Inheritance diagram for MOF_sim:



Public Member Functions

- **MOF_sim** (int n1=0, double p1=0)

  Set parametrs of the distribution with this constructor.
- **MOF_sim** (Distribution &d)

  Set the distribution itself with this constructor.
- void simulate (int volume) override

  .simulate(volume) writes "volume" realizations of random value of Bin(m,p) in "sample".

Additional Inherited Members

4.12.1 Detailed Description

A class inheriting SampleGenerator and overriding virtual simulating method with inverse function method. There is his brother, Ber_sim class that works faster with probabilities close to 0 or 1.

4.12.2 Member Function Documentation

4.12.2.1 simulate()

```
void MOF_sim::simulate (
            int volume )  [inline], [override], [virtual]
```

.simulate(volume) writes "volume" realizations of random value of Bin(m,p) in "sample".

Realizations will be modelled by inverse function method.

Implements SampleGenerator.

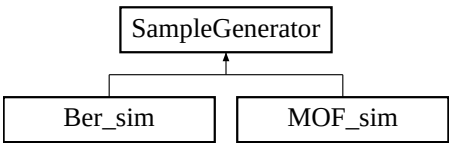
The documentation for this class was generated from the following file:

- MOF_sim.h

4.13 SampleGenerator Class Reference

```
#include <SampleGenerator.h>
```

Inheritance diagram for SampleGenerator:



Public Member Functions

- virtual void simulate (int volume)=0

  Method to be overridden by inherits.
- **SampleGenerator** (int n1=0, double p1=0.)

Public Attributes

- int * **sample**

Protected Attributes

- Distribution **distr**

4.13.1 Detailed Description

An abstract class containing Distribution and built for convenient polymorphism of the modelling methods.

4.13.2 Member Function Documentation

4.13.2.1 simulate()

```
virtual void SampleGenerator::simulate (
            int volume )  [pure virtual]
```

Method to be overridden by inherits.

.simulate(volume) writes "volume" realizations of random value of Bin(m,p) in "sample".

Implemented in Ber_sim, and MOF_sim.

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

- SampleGenerator.h

Chapter 5

File Documentation

5.1 Ber_sim.h

```
1 #pragma once
2 #include "R64M.hpp"
3 #include "SampleGenerator.h"
8 class Ber_sim :
9     public SampleGenerator
10 {
11 public:
13     Ber_sim(int n1 = 0, double p1 = 0) {
14         distr.set_m(n1);
15         distr.set_p(p1);
16         distr.probabilities();
17     }
19     Ber_sim(Distribution &d)
20     {
21         distr = d;
22         distr.probabilities();
23     }
26     void simulate(int volume) override
27     {
28         delete[] sample;
29         sample = new int[volume];
30         int k;
31         double alpha;
32         for (int j = 0; j < volume; ++j)
33         {
34             k = 0;
35             for (int i = 0; i < distr.get_m(); ++i)
36             {
37                 alpha = rnunif();
38                 k += int(trunc(alpha + distr.get_p()));
39             }
40             sample[j] = k;
41         }
42     }
43 };
44
```

5.2 Bin_Distr_Chi_Sq.h

```
1 #pragma once
5 typedef int (*Method)(int, double);
6
7 int Bin_MOF_NE(int m, double p);
8
9 int Bin_Ber_NE(int m, double p);
10
11 int Bin_MOF_VV(int m, double p);
12
13 int Bin_Ber_VV(int m, double p);
14
15 void Bin_Distr(int m, double p, double* a);
16
17 int Union(int n_distr, int n_sample, double* Exp, int* Freq, double* Out_pr, double* Out_fr);
18
19 int Count_p_vals(double alpha, double step, int n_p_values, double* p_values);
20
21 void P_value(int n_sample, int n_distr, double p, double& p_value, Method f, int n_dist_0, double p_0);
```

5.3 CFormatDlg.h

```
1 #pragma once
2
8 class CFormatDlg : public CDialog
9 {
10     DECLARE_DYNAMIC(CFormatDlg)
11
12 public:
13     CFormatDlg(CWnd* pParent = nullptr);
14     virtual ~CFormatDlg();
15
16 #ifdef AFX_DESIGN_TIME
17     enum { IDD = IDD_DIALOG1 };
18 #endif
19
20 protected:
21     virtual void DoDataExchange(CDataExchange* pDX);
22
23     DECLARE_MESSAGE_MAP()
24 public:
25     int m_n_par;
26     double m_p_par;
27     int m_vol_par;
28     int m_radio;
29 };
```

5.4 Chi_sq.h

```
1 #pragma once
2 #include "PROBDIST.h"
3 #include "Disrtibution.h"
4 #include "Bin_Distr_Chi_Sq.h"
10 class Chi_sq
11 {
12 public:
14     int n;
16     Distribution d;
18     int* sample;
19     double* ExpectedPr;
20     double* ExpectedFr;
22     int* hist;
24     double chi_sq;
25
27     int max_h;
29     double max_ex_fr;
30
31     double p_level;
32     int freedom_degree;
33
35     Chi_sq();
37     void Set_Data(int* sample1, int n1, Distribution &d0);
38
40     void Calc_chi_sq();
41
42     ~Chi_sq();
43 };
```

5.5 CplevelsDlg.h

```
1 #pragma once
2
8 class CplevelsDlg : public CDialog
9 {
10     DECLARE_DYNAMIC(CplevelsDlg)
11
12 public:
13     CplevelsDlg(CWnd* pParent = nullptr);
14     virtual ~CplevelsDlg();
15
16 #ifdef AFX_DESIGN_TIME
17     enum { IDD = IDD_DIALOG2 };
18 #endif
19
20 protected:
21     virtual void DoDataExchange(CDataExchange* pDX);
22
23     DECLARE_MESSAGE_MAP()
24 public:
```

```
25     int m_H0_m;
26     double m_H0_p;
27     int m_H1_m;
28     double m_H1_p;
29     int m_p_val_sam_size;
30     double m_alpha;
31     int m_n;
32     int m_meth;
33 };
```

5.6 CpowerDlg.h

```
1 #pragma once
2
7 class CpowerDlg : public CDialog
8 {
9     DECLARE_DYNAMIC(CpowerDlg)
10
11 public:
12     CpowerDlg(CWnd* pParent = nullptr);
13     virtual ~CpowerDlg();
14
15 #ifdef AFX_DESIGN_TIME
16     enum { IDD = IDD_DIALOG3 };
17 #endif
18
19 protected:
20     virtual void DoDataExchange(CDataExchange* pDX);
21
22     DECLARE_MESSAGE_MAP()
23 public:
24     int m_H0_m;
25     double m_H0_p;
26     int m_H1_m;
27     double m_H1_p;
28     double m_alpha;
29 };
```

5.7 Disrtibution.h

```
1 #pragma once
5 class Distribution
6 {
7     int m;
8     double p;
9 public:
10     int get_m() { return m; }
11     void set_m(int m1) { m = m1; this->probabilities();}
12     double get_p() { return p; }
13     void set_p(double p1) { p = p1; this->probabilities();}
14     double* expected_prob;
16     Distribution(int n1 = 1, double p1 = 0.1) : m(n1), p(p1), expected_prob(nullptr) {
       this->probabilities(); }
17     //Distribution(Distribution& d) = delete;
18     Distribution(Distribution *d) : m(d->get_m()), p(d->get_p()), expected_prob(nullptr) {
       this->probabilities(); }
19     ~Distribution(){delete[] expected_prob;}
23     void probabilities()
24     {
25         delete[] expected_prob;
26         expected_prob = new double[m + 1];
27         double r, c = p / (1. - p);
28         r = pow(1. - p, m);
29         expected_prob[0] = r;
30         for (int i = 1; i <= m; ++i)
31         {
32             r *= (c * (double(m) - double(i) + 1.) / i);
33             expected_prob[i] = r;
34         }
35     }
36 };
37
```

5.8 framework.h

```
1 #pragma once
```

```
2
3 #ifndef VC_EXTRALEAN
4 #define VC_EXTRALEAN
5 #endif
6
7 #include "targetver.h"
8
9 #define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS
10
11
12 #define _AFX_ALL_WARNINGS
13
14 #include <afxwin.h>
15 #include <afxext.h>
16
17
18
19
20
21 #ifndef _AFX_NO_OLE_SUPPORT
22 #include <afxdtctl.h>
23 #endif
24 #ifndef _AFX_NO_AFXCMN_SUPPORT
25 #include <afxcmn.h>
26 #endif
27
28 #include <afxcontrolbars.h>
29
30
31
32
33
34
35
36
37
38
39
```

5.9 Histogram.h

```
1
2 #pragma once
3
4 #ifndef __AFXWIN_H__
5     #error " pch.h      PCH"
6 #endif
7
8 #include "resource.h"
9
10 class CHistogramApp : public CWinAppEx
11 {
12 public:
13     CHistogramApp() noexcept;
14
15
16 public:
17     virtual BOOL InitInstance();
18
19     UINT  m_nAppLook;
20     BOOL  m_bHiColorIcons;
21
22     virtual void PreLoadState();
23     virtual void LoadCustomState();
24     virtual void SaveCustomState();
25
26     afx_msg void OnAppAbout();
27     DECLARE_MESSAGE_MAP()
28 };
29
30 extern CHistogramApp theApp;
```

5.10 HistogramDoc.h

```
1 #pragma once
2 #include "MOF_sim.h"
3 #include "Bin_Distr_Chi_Sq.h"
4 #include "Chi_sq.h"
5 #include "PROBDIST.h"
```

```
6 #include "Ber_sim.h"
7 #include "CFormatDlg.h"
8 #include "CplevelsDlg.h"
9 #include "CpowerDlg.h"
10 class CHistogramDoc : public CDocument
11 {
12 protected:
13     CHistogramDoc() noexcept;
14     DECLARE_DYNCREATE(CHistogramDoc)
15
16 public:
18     SampleGenerator* generator;
20     Distribution h0;
22     Distribution h1;
23
25     Chi_sq CHI_c;
26
28     int form_n;
30     int form_radio;
31
37     int cases;
38
40     double p_lev_alpha;
42     int p_lev_ss;
43
44     double* p_values;
46     double betas[12];
47 public:
48
49 public:
50     virtual BOOL OnNewDocument();
51     virtual void Serialize(CArchive& ar);
52 #ifdef SHARED_HANDLERS
53     virtual void InitializeSearchContent();
54     virtual void OnDrawThumbnail(CDC& dc, LPRECT lprcBounds);
55 #endif
56
57 public:
58     virtual ~CHistogramDoc();
59 #ifdef _DEBUG
60     virtual void AssertValid() const;
61     virtual void Dump(CDumpContext& dc) const;
62 #endif
63
64 protected:
65
66 protected:
67     DECLARE_MESSAGE_MAP()
68
69 #ifdef SHARED_HANDLERS
70     void SetSearchContent(const CString& value);
71 #endif
72 public:
73     afx_msg void OnFormat();
74     afx_msg void Onplevels();
75     afx_msg void On32775();
76 };
```

5.11 HistogramView.h

```
1 #pragma once
2
3
4 class CHistogramView : public CView
5 {
6 protected:
7     CHistogramView() noexcept;
8     DECLARE_DYNCREATE(CHistogramView)
9
10 public:
11     CHistogramDoc* GetDocument() const;
12
13 public:
14
15 public:
16     virtual void OnDraw(CDC* pDC);
17     virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
18 protected:
19
20 public:
21     virtual ~CHistogramView();
22 #ifdef _DEBUG
23     virtual void AssertValid() const;
```

```
24     virtual void Dump(CDumpContext& dc) const;
25 #endif
26
27 protected:
28
29 protected:
30     afx_msg void OnFilePrintPreview();
31     afx_msg void OnRButtonUp(UINT nFlags, CPoint point);
32     afx_msg void OnContextMenu(CWnd* pWnd, CPoint point);
33     DECLARE_MESSAGE_MAP()
34 };
35
36 #ifndef _DEBUG
37 inline CHistogramDoc* CHistogramView::GetDocument() const
38    { return reinterpret_cast<CHistogramDoc*>(m_pDocument); }
39 #endif
40
```

5.12 MainFrm.h

```
1
2
3 #pragma once
4
5 class CMainFrame : public CFrameWndEx
6 {
7
8 protected:
9     CMainFrame() noexcept;
10     DECLARE_DYNCREATE(CMainFrame)
11
12 public:
13
14 public:
15
16 public:
17     virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
18     virtual BOOL LoadFrame(UINT nIDResource, DWORD dwDefaultStyle = WS_OVERLAPPEDWINDOW | FWS_ADDTOTITLE,
       CWnd* pParentWnd = nullptr, CCreateContext* pContext = nullptr);
19
20 public:
21     virtual ~CMainFrame();
22 #ifdef _DEBUG
23     virtual void AssertValid() const;
24     virtual void Dump(CDumpContext& dc) const;
25 #endif
26
27 protected:
28     CMFCMenuBar       m_wndMenuBar;
29     CMFCToolBar       m_wndToolBar;
30     CMFCStatusBar     m_wndStatusBar;
31     CMFCToolBarImages m_UserImages;
32
33 protected:
34     afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
35     afx_msg void OnViewCustomize();
36     afx_msg LRESULT OnToolbarCreateNew(WPARAM wp, LPARAM lp);
37     afx_msg void OnApplicationLook(UINT id);
38     afx_msg void OnUpdateApplicationLook(CCmdUI* pCmdUI);
39     DECLARE_MESSAGE_MAP()
40
41 };
42
43
```

5.13 MOF_sim.h

```
1 #pragma once
2 #include "SampleGenerator.h"
3 #include "R64M.hpp"
8 class MOF_sim :
9     public SampleGenerator
10 {
11 public:
13     MOF_sim(int n1 = 0, double p1 = 0) {
14         distr.set_m(n1);
15         distr.set_p(p1);
16         distr.probabilities();
17     }
```

```
19      MOF_sim(Distribution &d)
20      {
21          distr = d;
22          distr.probabilities();
23      }
26      void simulate(int volume) override
27      {
28          delete[] sample;
29          sample = new int[volume];
30          double q, l, s, t;
31          int k;
32          if (distr.get_p() > 0.5)
33              q = 1. - distr.get_p();
34          else q = distr.get_p();
35          for (int j = 0; j < volume; ++j)
36          {
37              s = pow(1. - q, distr.get_m());
38              k = 0;
39              t = s;
40              l = q / (1. - q);
41
42              double alpha = rnunif();
43              while (alpha > s)
44              {
45                  ++k;
46                  t *= (l * (double(distr.get_m()) - k + 1.) / k);
47                  s += t;
48              }
49              if (distr.get_p() < 0.5)
50                  sample[j] = k;
51              else
52                  sample[j] = distr.get_m() - k;
53          }
54      }
55  };
```

5.14 pch.h

```
1
2  #ifndef PCH_H
3  #define PCH_H
4
5  #include "framework.h"
6
7  #endif
```

5.15 PROBDIST.H

```
1
2  void NORMAL( int type, double &x, double &p);
3  double pNormal(double x);
4  double xNormal(double prob);
5  void  CHI( int type, double n, double &x, double &p);
6  double pChi(double x, int n);
7  double xChi(double prob, int n);
```

5.16 R64M.hpp

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #pragma once
3
4
5  void  rninit   (unsigned long long iufir);
6  void  rnrest   ();
7  void  rnconst  (unsigned long long iufir);
8  //void  rnconfix (unsigned nmb);
9
10 unsigned long long rnfirst ();
11 unsigned long long rnlast  ();
12 //unsigned long rnconrd ();
13
14 double  rnunif ();
15 //double  rnexp  ();
16 //double  rnnorm ( );
```

5.17 resource.h

```
1 //{{NO_DEPENDENCIES}}
2
3 #define IDD_ABOUTBOX                    100
4 #define IDR_POPUP_EDIT                  119
5 #define ID_STATUSBAR_PANE1              120
6 #define ID_STATUSBAR_PANE2              121
7 #define IDS_STATUS_PANE1                122
8 #define IDS_STATUS_PANE2                123
9 #define IDS_TOOLBAR_STANDARD            124
10 #define IDS_TOOLBAR_CUSTOMIZE           125
11 #define ID_VIEW_CUSTOMIZE               126
12 #define IDR_MAINFRAME                   128
13 #define IDR_MAINFRAME_256               129
14 #define IDR_HistogramTYPE               130
15 #define IDR_THEME_MENU                  200
16 #define ID_SET_STYLE                    201
17 #define ID_VIEW_APPLOOK_WIN_2000        205
18 #define ID_VIEW_APPLOOK_OFF_XP          206
19 #define ID_VIEW_APPLOOK_WIN_XP          207
20 #define ID_VIEW_APPLOOK_OFF_2003        208
21 #define ID_VIEW_APPLOOK_VS_2005         209
22 #define ID_VIEW_APPLOOK_VS_2008         210
23 #define ID_VIEW_APPLOOK_OFF_2007_BLUE   215
24 #define ID_VIEW_APPLOOK_OFF_2007_BLACK  216
25 #define ID_VIEW_APPLOOK_OFF_2007_SILVER 217
26 #define ID_VIEW_APPLOOK_OFF_2007_AQUA   218
27 #define ID_VIEW_APPLOOK_WINDOWS_7       219
28 #define IDS_EDIT_MENU                   306
29 #define IDD_DIALOG1                     310
30 #define IDD_DIALOG2                     312
31 #define IDD_DIALOG3                     314
32 #define IDC_EDIT1                       1003
33 #define IDC_EDIT2                       1004
34 #define IDC_EDIT3                       1005
35 #define IDC_RADIO1                      1006
36 #define IDC_RADIO2                      1007
37 #define IDC_H1_p                        1008
38 #define IDC_p_val_sam_size              1009
39 #define IDC_H0_m                        1010
40 #define IDC_H0_p                        1011
41 #define IDC_H1_m                        1012
42 #define IDC_alpha                       1013
43 #define IDC_n                           1014
44 #define IDC_RADIO3                      1015
45 #define ID_32771                        32771
46 #define ID_FORMAT                       32772
47 #define ID_Menu                         32773
48 #define ID_p_levels                     32774
49 #define ID_32775                        32775
50 #define ID_BUTTON32778                  32778
51
52 // Next default values for new objects
53 //
54 #ifdef APSTUDIO_INVOKED
55 #ifndef APSTUDIO_READONLY_SYMBOLS
56 #define _APS_NEXT_RESOURCE_VALUE        316
57 #define _APS_NEXT_COMMAND_VALUE         32779
58 #define _APS_NEXT_CONTROL_VALUE         1016
59 #define _APS_NEXT_SYMED_VALUE           310
60 #endif
61 #endif
```

5.18 SampleGenerator.h

```
1 #pragma once
2 #include "R64M.hpp"
3 #include "Disrtibution.h"
7 class SampleGenerator
8 {
9 protected:
10     Distribution distr;
11 public:
12     int* sample;
15     virtual void simulate(int volume) = 0;
16     SampleGenerator(int n1 = 0, double p1 = 0.)
17     {
18         distr.set_m(n1);
19         distr.set_p(p1);
20         sample = nullptr;
21         //distr.expected_prob = nullptr;
22         distr.probabilities();
```

```
23     }
24     ~SampleGenerator() { delete[] sample; }
25 };
26
```

5.19 targetver.h

```
1 #pragma once
2
3 #include <SDKDDKVer.h>
```

Index