

8. Ī óñòü äáí ú n í áçààèñèì ú ò ñëó÷àéí ú ò ááèè÷èí ϕ_i , $i=1..n$, ðàñĩ ðáááèáí í ú ò í í ðì àèüí í ñ í àðàì áððàì è μ è σ . Ñóì ì à éáááððàðí á èò í ðééí í áí èé í ð ñðááí ááí àðèòì áðè÷áñéí áí, ááèáí í àý í à σ^2 , ðàñĩ ðáááèáí à èàè òè-éáááððàð ñ n-1 ñòáí áí ýì è ñáí áí áü. Õí æá ñàì í á í í áðóáí í ó. Áñèè \bar{X} è s^2 í í ðáááèèèòü ñòáí áàððóí ú ì í áðàçíì, òí $ns^2/\sigma^2 \sim \chi^2(x | n-1)$.

Ááí áðàòèý ñëó÷àéí ú ò ÷èñáè

Áñèè r_i ðàñĩ ðáááèáí ú ðááí í ì áðí í í à $[0,1]$, à ϕ í í á÷èí ýàòñý ñòáí áàððóí í í ó í í ðì àèüí í í ó ðàñĩ ðáááèáí èð, òí í ðè ÷áðí ú ò f ñëó÷àéí àý ááèè÷èí à ðàñĩ ðáááèáí à èàè òè-éáááððàð ñ f ñòáí á-í ýì è ñáí áí áü, à í ðè í á÷áðí ú ò f òàè ðàñĩ ðáááèáí à ááèè÷èí à $-\frac{1}{2} \log \left(\prod_{i=1}^{(f-1)/2} r_i \right) + \phi$. Ñëó÷àéí àý ááèè÷èí à ϕ^2 ðàñĩ ðáááèáí à èàè òè-éáááððàð ñ í áí í é ñòáí áí úð ñáí áí áü.

Áü ÷èñéáí èá ó óí èöèè ðàñĩ ðáááèáí èý è áá éàáí òèèáé

Ðàñĩ ðáááèáí èá òè-éáááððàð ñ f ñòáí áí ýì è ñáí áí áü ñí áí ááááð ñ ááì ì à-ðàñĩ ðáááèáí èáì ñ í àðà-ì áððóí ì àñø òàáá 2 è í àðàì áððóí ó í ðì ú f/2: $\chi^2(x/f) = \Gamma(x/f/2)$. Ýòí ñí òí í ø áí èá è èñ-í í èüçí ááí í á í ðèáí àèì ú ò í èæá éí áàð.

Òàéè chi2DF.h

```
#ifndef __CHI2_H__
#define ENTRY extern
#define LOCAL static

double chi2DF(double n, double x);
double inv_chi2DF(double n, double x);

#define __CHI2_H__
#endif /* Ends #ifndef __CHI2_H__ */
```

Òàéè chi2DF.cpp

```
#include <assert.h>
#include <math.h>

#include "gammaDF.h"

ENTRY double
chi2DF(double f, double x)
/*
 * Áü÷èñéýàòñý ááðíýðííñòü òíáí, ÷òí ñëó÷àéíàý ááèè÷èíà,
 * ííá÷èíýðüàýñý ðàñĩðáááèáíèð òè-éáááððàð ñ f ñòáíáíýìè ñáíáíáü,
 * íá íðááíñòíáèè çáááíííáí çíà÷áíèý x.
 */
{
    assert(f > 0);
```

```
n GammaDF(f/2.0).value(x/2.0);
}/*chi2DF*/

ENTRY double
inv_chi2DF(double f, double q)
/* Îñÿäðñý çíàáíèà ò äéý êîðîíâî chi2DF(f, x) ääíàð q.
*/
{
    assert(f > 0);
    return GammaDF(f/2.0).quantile(q);
}/*inv_chi2DF*/
```