# 誤差関数 *erf* とプロビット

SHIMURA Masato
JCD02773@nifty.ne.jp

2010 年 3 月 1 日

## 目次



**概要**

K.E.Iverson が D.Knuth 達 (G.P.K) の Concrete Mathematics Comparison を著わしたときに超幾何級数 (H.) が J に組み込まれた。E.Show が H. を使った統計関数を提供したが、これが最近 J の Mailing List で更に深められ、ADD-ON で提供されている。この erf 関数を用いて、group probit を作成した。

## 1 erf

### 1.1 J での最初の作業

include　最初に `require 'plot numeric trig'`

ADD ON　`addons/stats/distribs/normal.ijs` を読み込む

locale　normal.ijs には locale が設定されているので `erf_pdistribs_` のように用いる

## 1.2 erf 関数と normal.ijs での関数定義

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$
$$erfc(x) = 1 - erf(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$$

erf の定義

$erf(z) = 1 - erf(z) = \frac{2z}{\sqrt{\pi}} {}_1F_1\left(\frac{1}{2}; \frac{3}{2}; -z^2\right) = \frac{2ze^{-z^2}}{\sqrt{\pi}} {}_1F_1\left(1; \frac{3}{2}; z^2\right)$

H. *Hypergeometrics* 関数を用いて積分を計算している。

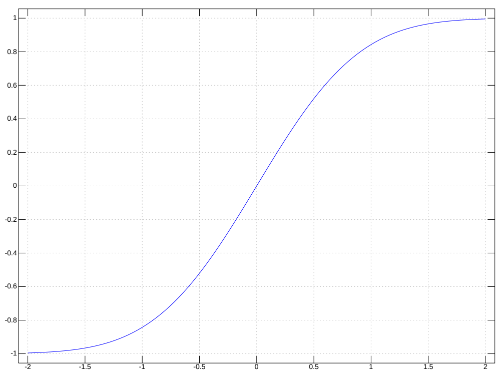
```
   %: 4p_1
1.12838
  2 %  %: 1p1
1.12838
```

```
NB. erf v error function
NB. ref Abramovitz and Stegum 7.1.21 (right)
erf=: (*&(%:4p_1)%^@:*:)*[:1 H. 1.5*:

NB. erfc v complementary error function
erfc=: >:@-@erf
```

$$\sqrt{\frac{4}{\pi}} = \frac{2}{\sqrt{\pi}}$$

## 1.3 erf を動かしてみる



```
 plot _2 2;'erf_pdistribs_'
```

_2 から 2 までを plot

```
  a,. erf_pdistribs_    a=. steps 0 1 10
  0        0
0.1 0.112463
0.2 0.222703
0.3 0.328627
0.4 0.428392
0.5   0.5205
0.6 0.603856
0.7 0.677801
0.8 0.742101
0.9 0.796908
  1 0.842701
```

### 1.4 正規分布関数を定義する

erf を用いた正規分布の定義

| | | |
|---|---|---|
| pnormh | $\Phi(x) = \frac{1}{2}\left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right)$<br>$= \frac{1}{2} erfc\left(-\frac{x}{\sqrt{2}}\right)$ | `NB. pnormh v Standard Normal CDF`<br>`NB. ref Abramovitz and Stegum 26.2.29 (solved for P)`<br>`pnormh=: (-: @: >: @ erf @ (%&(%:2))) f.` |

## 2 erf を用いた probit

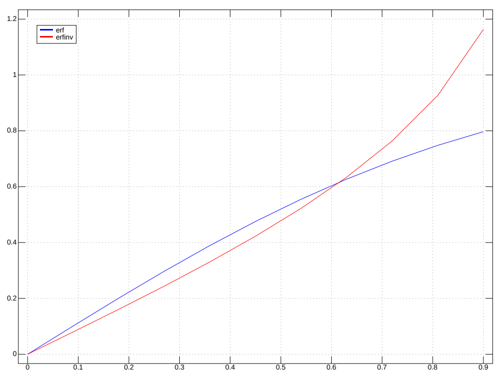
$$probit(p) = \Phi^{-1}(p) = \sqrt{2} erf^{-1}(2p - 1)$$

$erf^{-1}$ があれば $probit(x)$ は容易に定義できる。

*normal.ijs* の *qnorm* を用いた $erf^{-1}$ の定義は直接定義ファイルを見て欲しい。

```
NB. erfinv v inverse of error function
erfinv =: (0,%%:2) qnorm 0.5 + -:
```

```
 a=.steps 0 1 10
 a,. (erf_pdistribs_ a),.
 erfinv_pdistribs_ a
  0        0        0
0.1 0.112463 0.088856
0.2 0.222703 0.179143
0.3 0.328627 0.272463
0.4 0.428392 0.370807
0.5   0.5205 0.476936
0.6 0.603856 0.595116
0.7 0.677801 0.732869
0.8 0.742101 0.906194
0.9 0.796908  1.16309
  1 0.842701        _
```



■Worked Example　*(出典　Gujarati　table 15.4.)*

*D0*

*Xi* 所得階層 *(1000* ドル）

*Ni* 階層別構成所帯数

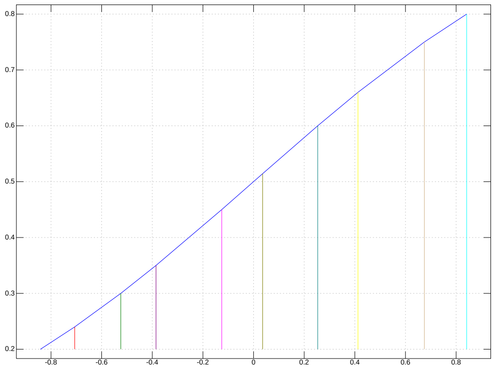
*ni* その階層での住宅保有所帯

```
   calc_probit_g D0
+-------------------------------------------------+
|Xi  Ni  ni     p    probit   Estim-nid  prob     |
+-------------------------------------------------+
| 6  40  8      0.2 _0.841621   _0.72478 0.234294|
| 8  50 12     0.24 _0.706303  _0.627847 0.265052|
|10  60 18      0.3 _0.524401  _0.530914 0.297739|
|13  80 28     0.35  _0.38532  _0.385515 0.349928|
|15 100 45     0.45 _0.125661  _0.288582 0.386451|
|20  70 36 0.514286 0.0358166 _0.0462501 0.481555|
|25  65 39      0.6  0.253347   0.196082 0.577727|
|30  50 33     0.66  0.412463   0.438414 0.669457|
|35  40 30     0.75   0.67449   0.680746 0.751984|
|40  25 20      0.8  0.841621   0.923078 0.822017|
+-------------------------------------------------+
```

| | | |
|---|---|---|
| *p* | $p = \frac{n_i}{N_i}$ | |
| *probit* | $probit(p) = \sqrt{2}erf^{-1}(2p-1)$ 分布曲線上の $p$ の $x$ 軸上の値を求める | `probit=:(%:@2:) * erfinv_pdistribs_@<:@+:` |
| *Estim-n.i.d* | *probit* の値を $X_i$ で回帰して推計値を求める。 | `find_rate=:   (%/)@(2 1 &{) "1`<br>`probit=:(%:@2:) * erfinv_pdistribs_@<:@+: NB. fork`<br>`NB. probit@find_rate D0`<br>`reg_probit=: probit@find_rate %. 1&,.@{."1 NB. OLS`<br>`estim_probit=:3 : '(reg_probit y)&p.{."1 y' NB.estimate`<br>`probit_main=:  pnormh_pdistribs_@estim_probit` |

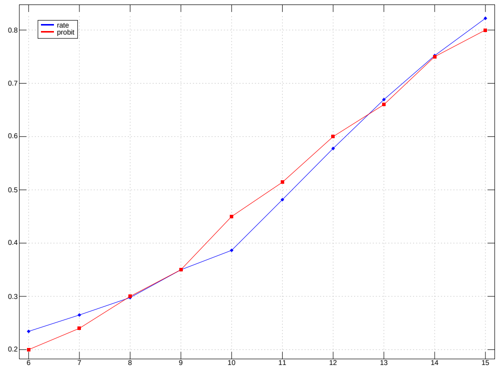
| | | |
|---|---|---|
| *proba-bility* | 回帰直線上の推計値を正規分布の確率変数の値に戻す | `pnormh_pdistribs_@estim_probit D0`<br>*normal.ijs* の関数<br>*p* の値が確率 *(prob)* に変換されており、階層間での変化も求められる。 |

```
(probit@find_rate;find_rate)L:0 D0
```



*p* の割合（累積密度関数）を x 軸上の値（逆関数）に変換する
当初の確率（比率）と probit の確率

```
'key rate probit;type line,marker'
plot (6+i.10);
(probit_main ,: find_rate) L:0 D0
```




# 3 References

*erf Wolfram MathWorld* が詳しい
*Damodor N.Gujarati [Basic Econometrics] 4th Edition Mcgraw-hill 2003*

# 4 Miscellance

*J Home page* `http://www.jsoftware.com`
*ADDON NET* に繋がった状態で　*RUN* → *Package Manager* から必要な *ADDON* を *DL* する。
単独で *DL* する場合は *jsofware.com* の *Wiki* よりバージョンを合わせて *DL* して解凍する。