```
capture program drop try
version 6
/*
Are Delays in Academic Publishing Necessary?
By DEREK LESLIE .  Program to generate Figure 3 data



1 = stand dev of z
we simulate for b

THIS CONTAINS THE CALCULATION FOR THE EXPECTED STANDARD OF ACCEPTED PAPERS
2 = critical prob of acceptance of marginal paper.  This is the implied G
3 =  prob acceptance standard.  This gives the implied zt go for top % of papers
*/
program define try
clear

set more 0

# delimit ;


di  %4s "b" "   "  %5s  "   prsub    "  %5s  "jtpr   "  %5s  "  acpr     "
%5s  "sucpr     " %5s  " qcrit    " %5s  "Ez    " %5s  " Exst"   "    " %5s  "
Exacc"
;


# delimit cr


local  i =1
while `i' >=0 {
scalar b= `i'
scalar varz=`1'^2
scalar varv = (varz/b) - varz
scalar sdv=sqrt(varv)
*scalar list b b1 b2
scalar varq= varv+varz
scalar vare=varz*(1-b)
scalar probz=`3'

* P  implied critical value of zt
scalar pp=invnorm(1-`3')
scalar zt=100+pp*sqrt(varz)


scalar ee=invnorm(1-`2')
*scalar list ee
scalar ecrit=ee*sqrt(vare)
*scalar list ecrit
scalar aa= 100*(1-b)
scalar qcrit= (zt -  aa  - ecrit)/b
scalar Ez=aa+b*qcrit
* probality that q>qcrit
scalar normq=(qcrit-100)/sqrt(varq)
scalar probq=normprob(normq)
scalar probq=1-probq

*prob z>zt &q>qcrit
```

```
scalar rhozq=sqrt(varz/varq)
scalar normz=pp
scalar jtprob=binorm(-normz, -normq, rhozq)
scalar accepp=jtprob/probq
scalar success= jtprob/probz

* Mean standard of ALL papers above zt standard. NOT THE MEAN OF ACCEPTED PAPERS
scalar sdz=sqrt(varz)
scalar alpha= (zt-100)/sdz
scalar norm = normd(alpha)
scalar mill = normd(alpha)/(1-normprob(alpha))
scalar expz=100 + (mill*sdz)
*SEE GREENE P707

* This is the expected standard of submitted papers to top journal
scalar sdq=sqrt(varq)
scalar alphq = (qcrit-100)/sdq
scalar millq = normd(alphq)/(1-normprob(alphq))
*scalar list millq
scalar expzq = 100+rhozq*millq*sdz
* HERE I AM ATTEMPTING THE EXPECTED STANDARD OF ACCEPTED PAPERS

* Now i calculate the variance
scalar xx = millq*(millq-alphq)

scalar varexpzq= varz*(1-(rhozq^2)*xx)
* NOW I CALCULATE THE EXPECTED VALUE if z>zt
scalar alpha= (zt-expzq)/sqrt(varexpzq)
scalar norm = normd(alpha)
scalar mill = normd(alpha)/(1-normprob(alpha))
scalar expz2=expzq + mill*sqrt(varexpzq)


* EXPERIMENT TO SEE IF IT WORKS EXPECTED STANDARD OF ACCEPTED PAPERS

*scalar list qcrit
*scalar list _all



*NUMERICAL METHOD - second best to check Mills stuff
/*
capture drop q
capture drop v
capture drop z
quietly set obs 10000000
gen z=100 + sdz*invnorm(uniform())
gen v= sdv*invnorm(uniform())
gen q=z+v
quietly su z if z>=zt&q>=qcrit
scalar exppub=r(mean)
quietly su z if q>=qcrit
scalar expp1=r(mean)

*/
scalar probq=100*probq
scalar jtprob=100*jtprob
scalar accepp=100*accepp
scalar success=100*success
scalar probz=100*probz
```

```
# delimit ;


 di %6.3f b " "  %6.4f  probq  "  "  %6.4f  jtprob "  "  %6.4f  accepp  " "
%6.4f  success "   "
 %6.2f  qcrit "  " %6.2f  Ez "   " %6.2f  expzq "    " %6.2f  expz2

;
# delimit cr


local i=`i' -.01

}

di %4s "zt"%4s "      critp "%4s "  aimprob"  %4s "  exp standZ>zt"


di %6.2f  zt "   "     %6.3f  100* `2'  "   "  %6.2f  probz  "  " %6.2f  expz




*su z if z>zt
*su z if q >=qcrit
*su z if z<zt
*regress z q
*/


end
```