BOSTON COLLEGE
Department of Economics
EC 771: Econometrics
Spring 2011
Prof. Baum, Mr. Smith

Problem Set 4: Solutions

Point Distribution:
1), 2), 3) : 10 points each

1)

```
. capt program drop ps41

. program ps41, rclass
  1. drop _all
  2. quietly set obs 500
  3. generate x = rt($df)
  4. summarize x, detail
  5. return scalar mu = r(mean)
  6. return scalar med = r(p50)
  7. return scalar df = $df
  8. end
.
. global nsim 1000

. global df 100

. scalar theor = c(pi)/2

. ps41

                              x
-------------------------------------------------------------
      Percentiles      Smallest
 1%    -2.178496       -3.30755
 5%    -1.660841      -2.662812
10%    -1.322605      -2.465379       Obs                 500
25%    -.7442768       -2.28337       Sum of Wgt.         500

50%     .0044037                      Mean            -.037566
                        Largest       Std. Dev.       .9865858
75%     .6558092       2.229074
```

```
90%     1.236853        2.24305       Variance       .9733515
95%     1.585368       2.838025       Skewness      -.0096165
99%     2.126817       3.050234       Kurtosis        2.79699

. set seed 20110406

. foreach df in 3 6 10 100 {
  2.         global df `df'
  3.         qui simulate mu = r(mu) med = r(med)  df = r(df), reps($nsim) nodots ///
>         saving(ps41_`df', replace): ps41
  4.
. // Calculate mean squares of median and mean, given both are unbiased estimators.
.         g double med2 = med^2
  5.         qui summ med2
  6.         scalar numer = r(mean)
  7.         g double mu2 = mu^2
  8.         qui summ mu2
  9.         scalar denom = r(mean)
 10.         scalar ratio = numer / denom
 11.         di _n "For `df' d.f.,  variance ratio = " ratio " vs. theoretical = " theor
 12. }

For 3 d.f.,  variance ratio = .63639509 vs. theoretical = 1.5707963

For 6 d.f.,  variance ratio = 1.1881244 vs. theoretical = 1.5707963

For 10 d.f.,  variance ratio = 1.3032442 vs. theoretical = 1.5707963

For 100 d.f.,  variance ratio = 1.6338293 vs. theoretical = 1.5707963
```

Note as the degrees of freedom increase, the variance ratio approaches the theoretical ratio of $\frac{\pi}{2} \approx 1.57$. This makes sense since the t-distribution approximates the normal distribution as the degrees of freedom increase.

2)

```
. prog ps42, rclass
1. drop _all
2. qui set obs 50
3. g x = rnormal()
4. g z = rnormal()
5. if ($case == 1) {
6.         g eps = rnormal()
7. }
8. else if ($case == 2) {
```

```
        g eps = rt(5)
}
else if ($case == 3) {
        g eps = exp(0.2 * x) * rnormal()
}
g y = 1 + x + $gamma * z + eps
reg y x z
scalar tgamma = _b[z] / _se[z]
return scalar rej = tgamma^2 > 3.84
end

. global nsim 1000

. global gamma 0.9

. global case 1

. ps42

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  2,    47) =   38.60
       Model |  74.3673906     2  37.1836953           Prob > F      =  0.0000
    Residual |  45.2785697    47  .963373823           R-squared     =  0.6216
-------------+------------------------------           Adj R-squared =  0.6055
       Total |   119.64596    49  2.44175429           Root MSE      =  .98152

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |    1.09329   .1450482     7.54   0.000     .8014903    1.385089
           z |     .65102    .154897     4.20   0.000     .3394075    .9626325
       _cons |   1.033639   .1426087     7.25   0.000     .7467476    1.320531
------------------------------------------------------------------------------

. ret li

scalars:
                r(rej) =  1

.
. mat res = J(12, 3, .)

. local gamma -0.5 0.2 0.6 0.9

. local i 0
```

```
. foreach g of local gamma {
  2.         forv c = 1/3 {
  3.                 global case `c'
  4.                 global gamma `g'
  5.                 local ++i
  6.                 mat res[`i',1] = `g'
  7.                 mat res[`i',2] = `c'
  8.                 qui simulate rej = r(rej), reps($nsim) nodots: ps42
  9.                 su rej, mean
 10.                 mat res[`i',3] = r(mean)
 11.         }
 12. }

. mat colnames res = Gamma EpsModel PrRej

. mat list res

res[12,3]
        Gamma  EpsModel     PrRej
 r1       -.5         1       .92
 r2       -.5         2      .749
 r3       -.5         3       .88
 r4        .2         1       .28
 r5        .2         2      .205
 r6        .2         3      .303
 r7        .6         1      .974
 r8        .6         2      .897
 r9        .6         3      .975
r10        .9         1         1
r11        .9         2      .987
r12        .9         3         1
```

Note that the probability of rejecting the null of $\gamma = 0$ using the Wald statistic increases as the distance of the true γ from zero increases. Also notice that the probability of rejection is lower when the error process is t-distributed. The distribution in this case has fatter tails and leads to less precise tests.

3) Note: You did not need to use Mata to do this problem.

```
  . capt prog drop ps43

. mata: mata clear

. mata:
------------------------------------------------- mata (type end to exit) -----------------
: void function lm(string scalar vname, string scalar vname2)
```

```
> {
>     vars = tokens(vname)
>     v = vars[|1,.|]
>     st_view(X,.,v)
>         vars = tokens(vname2)
>     v = vars[|1,.|]
>     st_view(e,.,v)
>         LM = (e' * X * invsym(X' * X) * X' * e) / ((e' * e) :/ rows(e))
>         st_numscalar("lm", LM)
> }
note: variable X may be used before set
note: variable e may be used before set

: end
------------------------------------------------------------------------------------------

. prog ps43, rclass
  1. drop _all
  2. qui set obs 50
  3. g x = rnormal()
  4. g z = rnormal()
  5. g iota = 1
  6. if ($case == 1) {
  7.         g eps = rnormal()
  8. }
  9. else if ($case == 2) {
 10.         g eps = rt(5)
 11. }
 12. else if ($case == 3) {
 13.         g eps = exp(0.2 * x) * rnormal()
 14. }
 15. g y = 1 + x + $gamma * z + eps
 16. // Run the wrong regression and get the residuals.
. reg y x
 17. predict double e, resid
 18. // Calculate the LM statistic for this regression.
. mata: lm("x z iota","e")
 19. ret scalar lm = lm
 20. return scalar rej = lm > 3.84
 21. end

. global nsim 1000

. global gamma 0.9

. global case 1
```

```
. ps43

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  1,    48) =   23.06
       Model |  38.9568037     1  38.9568037           Prob > F      =  0.0000
    Residual |  81.0804436    48  1.68917591           R-squared     =  0.3245
-------------+------------------------------           Adj R-squared =  0.3105
       Total |  120.037247    49  2.44973974           Root MSE      =  1.2997

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   .8544895   .1779314     4.80   0.000     .4967345    1.212244
       _cons |    1.12147   .1844813     6.08   0.000     .7505456    1.492395
------------------------------------------------------------------------------

. ret li

scalars:
                 r(rej) =  1
                  r(lm) =  15.15231059228623

.
. mat res = J(12, 3, .)

. local gamma -0.5 0.2 0.6 0.9

. local i 0

. foreach g of local gamma {
  2.         forv c = 1/3 {
  3.                 global case `c'
  4.                 global gamma `g'
  5.                 local ++i
  6.                 mat res[`i',1] = `g'
  7.                 mat res[`i',2] = `c'
  8.                 qui simulate rej = r(rej), reps($nsim) nodots: ps43
  9.                 su rej, mean
 10.                 mat res[`i',3] = r(mean)
 11.         }
 12. }

. mat colnames res = Gamma EpsModel PrRej

. mat list res
```

```
res[12,3]
          Gamma  EpsModel      PrRej
 r1         -.5         1       .918
 r2         -.5         2       .767
 r3         -.5         3       .886
 r4          .2         1       .296
 r5          .2         2       .208
 r6          .2         3       .286
 r7          .6         1       .976
 r8          .6         2       .872
 r9          .6         3       .975
r10          .9         1       .998
r11          .9         2       .985
r12          .9         3       .999
```

Similar to Question 2, the probability of rejecting the null of $\gamma = 0$ using the LM statistic increases as the distance of the true γ from zero increases. Note that we should not be very surprised that the results in Question 2 and Question 3 are similar as the two tests are equivalent asymptotically.