Statistics/Data Analysis

```
      name:  <unnamed>
       log:  C:\Users\Conor\Documents\Conor\Grad School\TA Work\Econ 103 - Econometric
> s\STATA Work\Week 6\wk6_section_log.smcl
  log type:  smcl
 opened on:   5 Feb 2018, 12:54:38

. 
. // Demonstration STATA code for week 6
. // Principles of Econometrics 4th Edition
. // Covered Problems: 5.15 and 6.22
. 
. set more off

. clear all

. use fair4.dta, clear

. 
. ////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> //////////////////////////////// Question 5.19 \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> ////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> 
. ******************************************************************************
. *Setup: Consider presidential voting data and consider a model:
. *
. * VOTE = beta1 + beta2*GROWTH + beta3*INFLATION + e
. *
. * Parts (A) - (C)
. ******************************************************************************
. 
. ******************************************************************************
. *Part A: Report the results of the regression in a standard format. Are the
. * estimates for beta2 and beta3 significantly different from zero at a 10%
. * confidence level? Did you use one-tail or two-tail tests? Why?
. ******************************************************************************
. 
. // Run regression
. reg vote growth inflation

      Source |       SS           df       MS      Number of obs   =        33
-------------+----------------------------------   F(2, 30)        =      8.11
       Model |  412.010088         2  206.005044   Prob > F        =    0.0015
    Residual |  761.840619        30  25.3946873   R-squared       =    0.3510
-------------+----------------------------------   Adj R-squared   =    0.3077
       Total |  1173.85071        32  36.6828346   Root MSE        =    5.0393

------------------------------------------------------------------------------
        vote |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      growth |   .6434205   .1656289     3.88   0.001     .3051611    .9816799
   inflation |  -.1720765   .4289553    -0.40   0.691     -1.04812    .7039672
       _cons |   52.15653   1.458703    35.76   0.000     49.17746     55.1356
------------------------------------------------------------------------------

. 
. // Store variance-covariance matrix
. matrix vcvEst = e(V)

. 
```

```
31. /* Discussion:
  >
  > To separately conduct a two-tailed test for the each of the two nulls that
  > (a) beta2 = 0 and (b) beta3 = 0 we can use the p-values provided in the STATA
  > output. Based on the reported results, at the 10% confidence level beta2 is
  > significantly different from zero, while beta3 is not. We can see this because
  > p-value(b2!=0) = 0.001 < 0.1 while p-value(b3!=) = 0.691 > 0.1
  >
  > Given the question of whether something is different from zero, we typically use
  > the two-sided test. This is equivalent to asking if the RHS variable - growth or
  > inflation - has any effect on the dependent variable (VOTE = incumbent vote share).
  >
  > Suppose you wanted to do 1-sided tests - we would expect b2 to be positive and
  > b3 to be negative, but you want to use the reported STATA output. How can we turn
  > the reported p-value into a 1-sided p-value?
  >
  > We know that for a 1-sided test, the p-value is one half that for the 2-sided test
  > when the t-stat has the correct sign for a rejection. Since we have the point
  > estimate b2 > 0, given the alternative that b2 > 0 we can calculate the p-value as:
  >
  > p-value(b2>0) = (1/2)*p-value(b2!=0)
  >
  > where != means "not equal". Similarly, since b3 < 0, we can calculate the p-value
  > for the alternative that b3 < 0 as:
  >
  > p-value(b3<0) = (1/2)*p-value(b3!=0)
  >
  > Using this approach, for the following tests we know:
  > (1) p-value(b2>0) is less than 0.001 so its less than 0.1, and we DO reject the
  >     null b2<= 0
  >
  > (2) p-value(b3<0) is about 0.3456, which is greater than 0.1 so we FAIL to
  >     reject the null that b3 >= 0
  >
  > */
32.
33. ********************************************************************************
34. *Part B: Assume the inflation rate is 4%. Predict the percentage vote for the
35. * incumbent party when growth is (i) -3%, (ii) 0%, and (iii) +3%.
36. ********************************************************************************
37.
38. // Before we begin, let's double check the units used for growth and inflation
39. sum growth inflation
```

Variable	Obs	Mean	Std. Dev.	Min	Max
growth	**33**	**.6242728**	**5.455028**	**-14.499**	**11.765**
inflation	**33**	**2.666303**	**2.106304**	**0**	**7.831**

```
40.
41. // We can see that the units for growth and inflation are whole numbers, so that
42. // 4% inflation means we need to use the factor 4 (not 0.04)
43.
44. // Calculation Option (1): Use lincom
45. lincom _b[_cons] + 4*_b[inflation] + (-3)*_b[growth]
```

(1) **- 3*growth + 4*inflation + _cons = 0**

vote	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**49.53796**	**1.158671**	**42.75**	**0.000**	**47.17164**	**51.90429**

```
lincom _b[_cons] + 4*_b[inflation]

 ( 1)  4*inflation + _cons = 0

------------------------------------------------------------------------------
        vote |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   51.46823   1.042963    49.35   0.000     49.33821    53.59824
------------------------------------------------------------------------------

lincom _b[_cons] + 4*_b[inflation] + 3*_b[growth]

 ( 1)  3*growth + 4*inflation + _cons = 0

------------------------------------------------------------------------------
        vote |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   53.39849   1.151877    46.36   0.000     51.04604    55.75093
------------------------------------------------------------------------------


// Calculation Option (2): Calculate point estimate for vote by hand
scalar voteHat_infl4_growthm3 = _b[_cons] + 4*_b[inflation] + (-3)*_b[growth]

scalar voteHat_infl4_growth0 = _b[_cons] + 4*_b[inflation]

scalar voteHat_infl4_growth3 = _b[_cons] + 4*_b[inflation] + 3*_b[growth]


disp "Point Estimates of incumbent vote share (VOTE) when..."
Point Estimates of incumbent vote share (VOTE) when...

disp "Growth = -3: " voteHat_infl4_growthm3
Growth = -3: 49.537965

disp "Growth = 0: " voteHat_infl4_growth0
Growth = 0: 51.468226

disp "Growth = 3: " voteHat_infl4_growth3
Growth = 3: 53.398488


// Calculation Option (3): Add observations to dataset, impose RHS values
// inflation = 4 and growth = -3, 0, 3, and calculate fitted value using predict

// Step (1): Add observations to dataset
scalar origNumObs = _N // capture number of observations in dataset

set obs `=_N+3' // expand number of observations in workspace by 3
number of observations (_N) was 33, now 36


// Step (2): Impose values for inflation and growth
replace inflation = 4 in `=origNumObs+1'/`=origNumObs+3' // set inflation = 4 in the
>  new observations 1 through 3
(3 real changes made)

replace growth = -3 in `=origNumObs+1' // set growth = -3 in 1st new observation
(1 real change made)
```

```
69. replace growth = 0 in `=origNumObs+2' // set growth = 0 in 2nd new observation
  (1 real change made)

70. replace growth = 3 in `=origNumObs+3' // set growth = 3 in 3rd new observation
  (1 real change made)

71.
72. // Step (3): Use predict to calculate fitted values
73. predict voteHat, xb

74.
75. /* STATA Technical Note:
  >
  > Even though the new entries for inflation and growth were not part of the
  > original regression, predict will still calculate the fitted values corresponding
  > to the values of inflation and growth in ALL rows currently in the dataset
  >
  > */
76.
77. // Have STATA report fitted values (voteHat) for the "new" observations
78. list voteHat inflation growth in `=origNumObs+1'/`=origNumObs+3'
```

	voteHat	inflat~n	growth
34.	**49.53796**	**4**	**-3**
35.	**51.46823**	**4**	**0**
36.	**53.39849**	**4**	**3**

```
79.
80. ****************************************************************************
81. *Part C: Test, as an alternative hypothesis, that the incumbent party will get
82. * the majority of the expected vote when the growth rate is (i) -3%, (ii) 0%,
83. * and (iii) +3%. Use a 1% level of significance. If you were the president
84. * seeking re-election, why might you set up each of these hypothesis as an
85. * alternative rather than a null hypothesis?
86. ****************************************************************************
87.
88. // Calculate t-critical value for a 1% RHS test
89.
90. scalar alpha = 0.01

91. scalar critical_value = invttail(e(df_r), alpha)

92. disp "Critical Value for the RHS test (H1: vote > 50) is " critical_value
```

Critical Value for the RHS test (H1: vote > 50) is 2.4572615

```
93.
94. // Calculation Option (1): Use lincom (std error will equal stdp)
95.
96. lincom _b[_cons] + 4*_b[inflation] + (-3)*_b[growth]-50
```

(1) **- 3*growth + 4*inflation + _cons = 50**

vote	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**-.4620352**	**1.158671**	**-0.40**	**0.693**	**-2.828357**	**1.904286**

```
lincom _b[_cons] + 4*_b[inflation] -50
```

(1) **4*inflation + _cons = 50**

vote	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**1.468226**	**1.042963**	**1.41**	**0.169**	**-.6617891**	**3.598242**

```
lincom _b[_cons] + 4*_b[inflation] + 3*_b[growth] - 50
```

(1) **3*growth + 4*inflation + _cons = 50**

vote	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**3.398488**	**1.151877**	**2.95**	**0.006**	**1.046042**	**5.750934**

```

// Calculation Option (2): Use predict stdp to calculate variance of fitted value
// voteHat (stdp) and variance of forecast (stdf). Then calculate t-stat by
// using generate (gen) for each of the two options.

predict voteHat_stdp, stdp

predict voteHat_stdf, stdf

gen tstat_50_stdp = (voteHat-50)/voteHat_stdp

gen tstat_50_stdf = (voteHat-50)/voteHat_stdf

list tstat_50_stdp tstat_50_stdf in `=origNumObs+1'/`=origNumObs+3'
```

	tstat_5~p	tstat_5~f
34.	**-.3987639**	**-.0893548**
35.	**1.407746**	**.2853082**
36.	**2.950392**	**.6574385**

```

disp "Critical Value for the RHS test (H1: vote > 50) is " critical_value
```

Critical Value for the RHS test (H1: vote > 50) is 2.4572615

```

// Calculate Option (3): calculate standard errors by hand using variance-covariance
// matrix from the regression.

// STDP is the variance of voteHat, or the variance of:
//
// b1 + G*b2 + 4*b3 where G can be -3, 0, or 3
//
// Thus, we can use the formula for standard error of a linear combination of
// variables. That is:
//
// Var(voteHat) = 1*var(b1) + 4^2*var(b3) + G^2*var(b2) +
//                  2*4*cov(b1,b3) + 2*G*cov(b1,b2) + 2*4*G*cov(b2,b3)
```

```
// and s.e.(voteHat) = sqrt(Var(voteHat))

scalar voteHat_stdp_infl4_growthm3 = sqrt( vcvEst[3,3] + (4^2)*vcvEst[2,2] + ((-3)^2
> )*vcvEst[1,1] + ///
>                                                    2*(1)*(4)*vcvEst[2,3] + 2*(1)*(-3)*vcvE
> st[1,3] + 2*(4)*(-3)*vcvEst[1,2] )

scalar voteHat_stdp_infl4_growth0 = sqrt( vcvEst[3,3] + (4^2)*vcvEst[2,2] + ///
>                                                    2*(1)*(4)*vcvEst[2,3] )

scalar voteHat_stdp_infl4_growth3 = sqrt( vcvEst[3,3] + (4^2)*vcvEst[2,2] + (3^2)*vc
> vEst[1,1] + ///
>                                                    2*(1)*(4)*vcvEst[2,3] + 2*(1)*(3)*vcvEs
> t[1,3] + 2*(4)*(3)*vcvEst[1,2] )


// STDF is the variance of the forecast error, or Var(vote - voteHat) which
// turns out to equal Var(voteHat) + Var(ei) where ei is a single error term
// Thus, s.e.(forecast) = sqrt( Var(voteHat) + sigma_hat^2 ) and we can use
// STATA's stored value e(rmse) to plug in the estimate of sigma_hat

scalar voteHat_stdf_infl4_growthm3 = sqrt(voteHat_stdp_infl4_growthm3^2 + e(rmse)^2)

scalar voteHat_stdf_infl4_growth0 = sqrt(voteHat_stdp_infl4_growth0^2 + e(rmse)^2)

scalar voteHat_stdf_infl4_growth3 = sqrt(voteHat_stdp_infl4_growth3^2 + e(rmse)^2)


// Calculate t-statistics using stdp

scalar tstat_50_stdp_infl4_growthm3 = (voteHat_infl4_growthm3 - 50)/voteHat_stdp_inf
> l4_growthm3

scalar tstat_50_stdp_infl4_growth0 = (voteHat_infl4_growth0 - 50)/voteHat_stdp_infl4
> _growth0

scalar tstat_50_stdp_infl4_growth3 = (voteHat_infl4_growth3 - 50)/voteHat_stdp_infl4
> _growth3


// Calculate t-statistics using stdf

scalar tstat_50_stdf_infl4_growthm3 = (voteHat_infl4_growthm3 - 50)/voteHat_stdf_inf
> l4_growthm3

scalar tstat_50_stdf_infl4_growth0 = (voteHat_infl4_growth0 - 50)/voteHat_stdf_infl4
> _growth0

scalar tstat_50_stdf_infl4_growth3 = (voteHat_infl4_growth3 - 50)/voteHat_stdf_infl4
> _growth3


/* Discussion:
>
> The null that vote <= 50 is the null of losing (or tying), while the alternative
> of vote>50 is winning. Setting up the desired outcome (vote>50 = winning) as the
> alternative means that you need strong evidence to support an expectation of
> winning while the default position is that you will lose. This sets up your
> reject/not reject as a high bar for thinking you will win and thus is a
> conservative way to decide if you think you're likely to win.
>
> Using the STDP standard errors, the t-stat for INFLATION = 4 and GROWTH = -3 is
> negative (point estimate is less than 50), so we cannot reject the null the
> E(vote)<50. For INFLATION = 4 and GROWTH = 0, the point estimate is positive,
> but not large enough for us to reject the null of E(vote)<50 since the t-stat
> is 1.41 versus the critical value of 2.45. Finally, for INFLATION = 4 and
> GROWTH = 3, we have a t-stat of 2.95 so that we can reject the null that
> E(vote)<50 at the 99% confidence level.
>
> Using the STDF standard errors, however, we get very small t-values and in all
> three cases we cannot reject the null. This is because we have a large estimate
> for sigma_hat^2 in this regression, reflected in part in the modest R2 of 0.35.
```

```
>
> Comment on STDP (variance of prediction) versus STDF (variance of forecast)
>
> The STDP is the variance of our expected value for vote. Thus, rejecting the null
> using the STDP variance is rejecting the null that the AVERAGE or EXPECTED outcome
> is a loss. However, there is still significant remaining uncertainty around the
> REALIZED outcome relative to the EXPECTED outcome, over and above the uncertainty
> coming from the imprecision in our beta estimates. By definition, STDF > STDP
> so rejecting using the STDF standard error will always be harder than using the
> STDP standard errors.
>
> */

////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> ///////////////////////////////// Question 6.22 \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> ////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
>
********************************************************************************
*Setup: Evaluate the relationship between income, age, and spending on pizza
* Parts (A) - (B) use model:
*
* PIZZA = beta1 + beta2*AGE + beta3*INCOME + beta4*(AGE x INCOME) + e
*
* Parts (C) - (E) use model:
*
* PIZZA = beta1 + beta2*AGE + beta3*INCOME + beta4*(AGE x INCOME) +
*           beta5*(AGE^2 x INCOME) + e
*
* Part (F) also involves a (AGE^3 x INCOME) estimate
*
* Parts (A) - (F)
********************************************************************************

clear // will keep scalars and matricies from 5.19, but remove dataset

use pizza4.dta, clear


********************************************************************************
*Part A: Test the hypothesis that age does not affect pizza expenditure - that
* is, test the joint hypothesis H0: beta2=0, beta4=0. What do you conclude?
********************************************************************************

gen age_income = age*income

reg pizza age income age_income

      Source |       SS           df       MS      Number of obs   =        40
-------------+----------------------------------   F(3, 36)        =      7.59
       Model |  367043.248         3  122347.749   Prob > F        =    0.0005
    Residual |  580608.652        36  16128.0181   R-squared       =    0.3873
-------------+----------------------------------   Adj R-squared   =    0.3363
       Total |    947651.9        39  24298.7667   Root MSE        =       127

------------------------------------------------------------------------------
       pizza |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  -2.977423   3.352101    -0.89   0.380    -9.775799    3.820952
      income |   6.979905   2.822768     2.47   0.018     1.255067    12.70474
  age_income |  -.1232393   .0667187    -1.85   0.073    -.2585512    .0120725
       _cons |   161.4654   120.6634     1.34   0.189     -83.2513    406.1822
------------------------------------------------------------------------------
```

```

// Have STATA conduct the joint hypothesis test using the test command
test (age=0) (age_income=0)

 ( 1)  age = 0
 ( 2)  age_income = 0

       F(  2,    36) =    7.40
            Prob > F =    0.0020


// Given p-value of 0.002 we can reject the null that beta2=0 and beta4=0 in
// in favor of the alternative that at least one of beta2 and beta4 is non-zero.
// and thus age has some effect on pizza expenditures

// Can also calculate the test statistic by hand:

// Part (1): Collect sum of squared errors and degrees of freedom in
// unrestricted model
scalar sse_u = e(rss) // sum of squared errors - unrestricted model

scalar df_u = e(df_r) // unrestricted model degrees of freedom


// Part (2): Run restricted regression, and collect sum of squared errors
// restricted regression: impose beta2=0 beta4=0 and re-run OLS
qui reg pizza income

scalar sse_r = e(rss) // sum of squared errors - restricted model


/* Stata Technical Note:
>
> the qui in front of the reg command means that STATA
> will NOT report the regression table in the output window/log file. This is fine
> given that we only want to get the sum of squared errors in order to use it for
> a later calculation.
>
> */

/* Comment:
>
> Note that the restricted regression is NOT the same as changing the betas we
> are testing in the unresricted regression. Imposing the restriction changes the
> point estimate for ALL the betas in the model.
>
> */

// Part (3): Calculate F-statistic
// F-stat = ((sse_r-sse_u)/J)/(sse_u/df_u)
// where J = # of restrictions
scalar fstat = ((sse_r-sse_u)/2)/(sse_u/df_u)


// Part (4): Hypothesis testing
// Use J = # of restrictions, and df_u = degrees of freedom for F-distribution
scalar alpha = 0.05

scalar ftest_critical_05 = invF(2,df_u,1-alpha)
```

```
scalar ftest_pval = Ftail(2,df_u,fstat)

disp "F-Test for H0: beta2=0 and beta4=0: F-stat " fstat " v.s critical value " ftes
> t_critical_05 " (p-value = " ftest_pval ")"
F-Test for H0: beta2=0 and beta4=0: F-stat 7.3994582 v.s critical value 3.2594463 (p-v
> alue = .00203266)

*****************************************************************************
*Part B: Construct point estimates and 95% interval estimates of the marginal
* propensity to spend on pizza for individuals of age 20, 30, 40, 50, and 55.
* Comment on these estimates.
*****************************************************************************

// The "marginal propensity to spend" refers to how much of an additional dollar
// of income is spent on a given good. Thus, we are looking at the derivative
// of PIZZA with respect to INCOME. This gives us:
//
// Marginal Propensity to Spend = MPS(INCOME, AGE) = beta3 + beta4*AGE
//
// Given that beta4 (the beta for AGE x INCOME) is negative, the MPS will
// decline as age increases.

// Re-run regression using interaction syntax so we can use margins command
reg pizza age income c.age#c.income
```

Source	SS	df	MS
Model	367043.25	3	122347.75
Residual	580608.65	36	16128.0181
Total	947651.9	39	24298.7667

Number of obs	=	40
F(3, 36)	=	7.59
Prob > F	=	0.0005
R-squared	=	0.3873
Adj R-squared	=	0.3363
Root MSE	=	127

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
age	-2.977423	3.352101	-0.89	0.380	-9.775799	3.820952
income	6.979905	2.822768	2.47	0.018	1.255067	12.70474
c.age#c.income	-.1232394	.0667187	-1.85	0.073	-.2585512	.0120725
_cons	161.4654	120.6634	1.34	0.189	-83.25131	406.1822

```

// Calculation option (1): lincom (using a loop)
local agesToLoop 20 30 40 50 55

foreach x of local agesToLoop {
  2.         disp "Marginal Propensity to Spend on Pizza at AGE=" `x'
  3.
        lincom _b[income]+`x'*_b[c.age#c.income]
  4.
}
Marginal Propensity to Spend on Pizza at AGE=20

 ( 1)  income + 20*c.age#c.income = 0
```

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	4.515118	1.520394	2.97	0.005	1.431615	7.598621

```
Marginal Propensity to Spend on Pizza at AGE=30

 ( 1)  income + 30*c.age#c.income = 0
```

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**3.282725**	**.9048794**	**3.63**	**0.001**	**1.447544**	**5.117905**

Marginal Propensity to Spend on Pizza at AGE=40

(1) **income + 40*c.age#c.income = 0**

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**2.050331**	**.4650721**	**4.41**	**0.000**	**1.107121**	**2.993541**

Marginal Propensity to Spend on Pizza at AGE=50

(1) **income + 50*c.age#c.income = 0**

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**.8179375**	**.7099684**	**1.15**	**0.257**	**-.6219452**	**2.25782**

Marginal Propensity to Spend on Pizza at AGE=55

(1) **income + 55*c.age#c.income = 0**

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**.2017408**	**.9908536**	**0.20**	**0.840**	**-1.807804**	**2.211285**

```
// Calculation option (2): margins command
margins, dydx(income) at(age=(20 30 40 50 55))
```

Average marginal effects Number of obs = **40**
Model VCE : **OLS**

Expression : **Linear prediction, predict()**
dy/dx w.r.t. : **income**

1._at : age = **20**

2._at : age = **30**

3._at : age = **40**

4._at : age = **50**

5._at : age = **55**

	dy/dx	Delta-method Std. Err.	t	P>|t|	[95% Conf.	Interval]
income						
_at						
1	**4.515118**	**1.520394**	**2.97**	**0.005**	**1.431615**	**7.598621**
2	**3.282725**	**.9048794**	**3.63**	**0.001**	**1.447544**	**5.117905**
3	**2.050331**	**.4650721**	**4.41**	**0.000**	**1.107121**	**2.993541**
4	**.8179375**	**.7099684**	**1.15**	**0.257**	**-.6219452**	**2.25782**
5	**.2017408**	**.9908536**	**0.20**	**0.840**	**-1.807804**	**2.211285**

```

// Calculation option (3): compute by hand use beta and variance-covariance matrix
matrix vcvEst = e(V)

scalar alpha = 0.05

scalar tcritical = invt(e(df_r),1-alpha/2)

foreach x of local agesToLoop {
  2.         scalar mps_age`x' = _b[income] + `x'*_b[c.age#c.income]
  3.         matrix mps_var_age`x' = vcvEst["income","income"] + (`x')^2*vcvEst["c.age
> #c.income","c.age#c.income"] + 2*`x'*vcvEst["income","c.age#c.income"]
  4.         scalar mps_se_age`x' = sqrt(mps_var_age`x'[1,1])
  5.         disp "MPS at Age `x': " mps_age`x' " --> Confidence Interval: [" mps_age`
> x' - tcritical*mps_se_age`x' ", " mps_age`x' + tcritical*mps_se_age`x' "]"
  6. }
MPS at Age 20: 4.515118 --> Confidence Interval: [1.4316153, 7.5986208]
MPS at Age 30: 3.2827245 --> Confidence Interval: [1.4475441, 5.117905]
MPS at Age 40: 2.050331 --> Confidence Interval: [1.1071211, 2.993541]
MPS at Age 50: .81793751 --> Confidence Interval: [-.6219452, 2.2578202]
MPS at Age 55: .20174076 --> Confidence Interval: [-1.8078035, 2.211285]


********************************************************************************
*Part C: Modify the equation to permit a "life-cycle" effect in which the
* marginal effect of income on pizza expenditure increases with age, up to a
* point, and then falls. Do so by adding (AGE^2 x INCOME) to the model. What
* sign do you anticipate on this term? Estimate the model and test the
* significance of the coefficient for this variable. Did the estimate have the
* expected sign?
********************************************************************************

gen agesqr = age^2

gen agesqr_income = agesqr*income

// Run regression with constructed variables
reg pizza age income age_income agesqr_income

      Source |       SS           df       MS      Number of obs   =        40
-------------+----------------------------------   F(4, 35)        =      5.83
       Model |  378782.696         4  94695.6741   Prob > F        =    0.0011
    Residual |  568869.204        35  16253.4058   R-squared       =    0.3997
-------------+----------------------------------   Adj R-squared   =    0.3311
       Total |    947651.9        39  24298.7667   Root MSE        =    127.49

-------------------------------------------------------------------------------
        pizza |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
--------------+----------------------------------------------------------------
          age |  -2.038273   3.541904    -0.58   0.569     -9.22872    5.152173
       income |   14.09616   8.839862     1.59   0.120    -3.849713    32.04203
   age_income |  -.4703705   .4139079    -1.14   0.264    -1.310648    .3699071
agesqr_income |   .0042048   .0049476     0.85   0.401    -.0058393    .0142488
        _cons |   109.7208   135.5725     0.81   0.424     -165.506    384.9475
-------------------------------------------------------------------------------

matrix vcvEstAgeSqr = e(V)


// Run regression with interaction notation
```

```
reg pizza age income c.age#c.income c.age#c.age#c.income
```

Source	SS	df	MS
Model	378782.705	4	94695.6762
Residual	568869.195	35	16253.4056
Total	947651.9	39	24298.7667

Number of obs	=	40
F(4, 35)	=	5.83
Prob > F	=	0.0011
R-squared	=	0.3997
Adj R-squared	=	0.3311
Root MSE	=	127.49

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
age	-2.038273	3.541904	-0.58	0.569	-9.22872	5.152174
income	14.09616	8.839862	1.59	0.120	-3.849711	32.04204
c.age#c.income	-.4703706	.4139079	-1.14	0.264	-1.310648	.369907
c.age#c.age#c.income	.0042048	.0049476	0.85	0.401	-.0058393	.0142488
_cons	109.7208	135.5725	0.81	0.424	-165.506	384.9475

```

// Given the set up for the question, we would expect the coefficient on the
// (AGE^2 x INCOME) term to be negative. The alternative is that the term is
// positive. This means we want to do a RHS test.
scalar tstat = _b[c.age#c.age#c.income]/_se[c.age#c.age#c.income]

scalar alpha = 0.05

scalar tcritical = invt(e(df_r),1-alpha)

scalar pvalue = 1-t(e(df_r),tstat)


disp "Test that agesqr_income beta is negative (H0) vs. positive (H1):"
```

Test that agesqr_income beta is negative (H0) vs. positive (H1):

```
disp "T-stat " tstat " v.s. critical value " tcritical " (p-value = " pvalue ")"
```

T-stat .84986859 v.s. critical value 1.6895725 (p-value = .20058724)

```

/* Discussion:
>
> As mentioned earlier, we expected to get a negative coefficient. Instead, we
> calculated a positive beta for the (AGE^2 x INCOME) term. However, we fail to
> reject the null that the beta is negative. In addition, we would fail to reject
> the null that the beta = 0 in a two-sided test.
>
> */

********************************************************************************
*Part D: Using the model in (c), construct point estimates and 95% interval
* estimates of the marginal propensity to spend on pizza for individuals of ages
* 20, 30, 40, 50, and 55. Comment on these estimates. In light of these values,
* and of the range of age in the sample data, what can you say about the
* quadratic function of age that describes the marginal propensity to spend on
* pizza?
********************************************************************************

```

```
// With the new model, the marginal propensity to spend on pizza is:
// d PIZZA/d INCOME = beta3 + beta4*AGE + beta5*AGE^2
//
// The variance of this estimate will be:
// var(b3) + AGE^2*var(b4)+AGE^4*var(b5) +
//   2*AGE*cov(b3,b4) + 2*AGE^2*cov(b3,b5) + 2*AGE^3*cov(b4,b5)

// Calculation option (1): margins
margins, dydx(income) at(age=(20 30 40 50 55))
```

```
Average marginal effects                        Number of obs     =         40
Model VCE    : OLS

Expression   : Linear prediction, predict()
dy/dx w.r.t. : income

1._at        : age             =          20

2._at        : age             =          30

3._at        : age             =          40

4._at        : age             =          50

5._at        : age             =          55
```

	dy/dx	Delta-method Std. Err.	t	P>|t|	[95% Conf.	Interval]
income						
_at						
1	**6.370658**	**2.663923**	**2.39**	**0.022**	**.962608**	**11.77871**
2	**3.769337**	**1.073784**	**3.51**	**0.001**	**1.589439**	**5.949235**
3	**2.008969**	**.4694063**	**4.28**	**0.000**	**1.056024**	**2.961915**
4	**1.089555**	**.7811004**	**1.39**	**0.172**	**-.496163**	**2.675273**
5	**.9452059**	**1.324651**	**0.71**	**0.480**	**-1.743978**	**3.63439**

```

// Calculation option (2): lincom
local agesToLoop 20 30 40 50 55

foreach x of local agesToLoop {
  2.         disp "Marginal Propensity to Spend on Pizza - AGE = " `x'
  3.         scalar xsqr = `x'^2
  4.         lincom _b[income] + (`x')*_b[c.age#c.income] + (xsqr)*_b[c.age#c.age#c.in
> come]
  5. }
Marginal Propensity to Spend on Pizza - AGE = 20

 ( 1)  income + 20*c.age#c.income + 400*c.age#c.age#c.income = 0
```

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**6.370658**	**2.663923**	**2.39**	**0.022**	**.962608**	**11.77871**

```
Marginal Propensity to Spend on Pizza - AGE = 30

 ( 1)  income + 30*c.age#c.income + 900*c.age#c.age#c.income = 0
```

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**3.769337**	**1.073784**	**3.51**	**0.001**	**1.589439**	**5.949235**

```
Marginal Propensity to Spend on Pizza - AGE = 40

 ( 1)  income + 40*c.age#c.income + 1600*c.age#c.age#c.income = 0
```

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**2.008969**	**.4694063**	**4.28**	**0.000**	**1.056024**	**2.961915**

Marginal Propensity to Spend on Pizza - AGE = 50

(1) **income + 50*c.age#c.income + 2500*c.age#c.age#c.income = 0**

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**1.089555**	**.7811004**	**1.39**	**0.172**	**-.496163**	**2.675273**

Marginal Propensity to Spend on Pizza - AGE = 55

(1) **income + 55*c.age#c.income + 3025*c.age#c.age#c.income = 0**

pizza	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**.9452059**	**1.324651**	**0.71**	**0.480**	**-1.743978**	**3.63439**

```
// Calculation option (3): by hand
scalar alpha = 0.05

scalar tc_2side = invt(e(df_r),1-alpha/2)

foreach x of local agesToLoop {
  2.         scalar mps_m2_age`x' = _b[income] + `x'*_b[c.age#c.income] + `x'^2*_b[c.a
> ge#c.income]
  3.         scalar mps_m2_se_age`x' = sqrt( vcvEstAgeSqr[3-1,3-1] + (`x'^2)*vcvEstAge
> Sqr[4-1,4-1] + (`x'^4)*vcvEstAgeSqr[5-1,5-1] + ///
>                                                                  2*`x'*vcvEstAgeSqr[3
> -1,4-1] + 2*(`x'^2)*vcvEstAgeSqr[3-1,5-1] + 2*(`x'^3)*vcvEstAgeSqr[4-1,5-1] )
  4.         disp "MPS at Age `x': " mps_m2_age`x' " Confidence Interval: [" ///
>                         mps_m2_age`x' - tc_2side*mps_m2_se_age`x' ", " ///
>                         mps_m2_age`x' + tc_2side*mps_m2_se_age`x' "]"
  5. }
MPS at Age 20: 6.3706583 Confidence Interval: [.96260812, 11.778708]
MPS at Age 30: 3.7693368 Confidence Interval: [1.5894386, 5.949235]
MPS at Age 40: 2.0089691 Confidence Interval: [1.0560237, 2.9619145]
MPS at Age 50: 1.0895552 Confidence Interval: [-.49616299, 2.6752734]
MPS at Age 55: .94520593 Confidence Interval: [-1.7439777, 3.6343896]



*******************************************************************************
*Part E: For the model in part (c), are each of the coefficients estimates for
* AGE, (AGE x INC), and (AGE^2 x INC) significantly different from zero at a
* 5% significance level? Carry out a joint test for the significance of these
* variables. Comment on your results.
*******************************************************************************

// Calculation option (1): Use test command
test (_b[age]=0) (_b[c.age#c.income]=0) (_b[c.age#c.age#c.income]=0)

 ( 1)  age = 0
 ( 2)  c.age#c.income = 0
 ( 3)  c.age#c.age#c.income = 0

       F(  3,    35) =    5.14
            Prob > F =    0.0048
```

```
/* Discussion:
>
> Based on the results of our test, we can reject the null that all three betas
> are equal to zero. To be clear, the interpretation of this rejection is that
> AT LEAST ONE of the proposed restrictions is not valid. However, we cannot say
> which one is not valid, in what direction, or how many of the restrictions are
> not valid.
>
> */

// Calculation option (2): Run restricted regression, and calculate using:
// F = ((sse_r - sse_u)/J)/(sse_u/df_u)
scalar sse_u = e(rss)

scalar df_u = e(df_r)

qui reg pizza income

scalar sse_r = e(rss)

scalar fstat = ((sse_r - sse_u)/3)/(sse_u/df_u)


scalar alpha = 0.05

scalar ftest_cval = invF(3,df_u,1-alpha)

scalar fstat_pval = 1-F(3,df_u,fstat)


disp "Joint Test that all AGE betas = 0: F-Stat = " fstat ///
>                " v.s. 95% critical value " ftest_cval " (p-value = " fstat_pval ")"
```

Joint Test that all AGE betas = 0: F-Stat = 5.1356754 v.s. 95% critical value 2.874187
> 5 (p-value = .00476349)

```

******************************************************************************
*Part F: Check the model used in part (c) for collinearity. Add the term
* (AGE^3 x INCOME) to the model in (c) and check the resulting model for
* collinearity.
******************************************************************************

// Two ways to test for colinearity:
// (1) look at pair-wise correlations of variables
// (2) see how much of variation in one RHS variable is explained by all the
//      other RHS variables by running reg RHS1 RHS2 RHS3 RHS4 etc. and checking
//      the R^2 for each regression
//  --> Method (1) is quick and simple, but may miss potential interactions
//        among multiple variables that will be picked up by method (2)

// Since method (2) is more involved, let's calculate those values first and
// then report them along with the pair-wise correlations

// Create the new variable for AGE^3 x INCOME
gen agecube_income = age^3*income


// Create a local varlist with the names of the variables we'll be checking
local varsToLoop income age age_income agesqr_income agecube_income
```

```

// Create a local that has the number of items in varsToLoop
local numVars : word count `varsToLoop'

// Prepare the matrix chkCL that will store R^2 values generated in the loop
// Initiate the matrix with empty values using .
matrix chkCL = J(`numVars',2,.)

// give chkCL column names and rownames
matrix colnames chkCL = "ptC" "ptF"
matrix rownames chkCL = `varsToLoop'

// Create a "count" variable to see where in the loop we are
scalar count = 0

foreach x of local varsToLoop {
  2.          // Update count for how which iteration of the loop we're in
        scalar count = count+1
  3.
        // select only the current variable, place name in local tmpLHS
        local tmpLHS `x'
  4.
        // for RHS group 1, exclude current LHS and agecube_income
        // (since agecube_income isn't from part c)
        local excl1 agecube_income `tmpLHS'
  5.
        // select RHS from part (c) other than current variable being tested
        // Equivalent to all listed variables EXCLUDING whats those listed in excl1
        local tmpRHS1 : list varsToLoop -excl1
  6.
        // for RHS groups 2, only exclude current LHS variable
        local tmpRHS2 : list varsToLoop -tmpLHS
  7.
        // Regress tmpLHS variable on tmpRHS1 variables, store R^2 in column 1
        qui reg `tmpLHS' `tmpRHS1'
  8.          matrix chkCL[count,1] = e(r2)
  9.
        // Regress tmpLHS variable on tmpRHS2 variables, store R^2 in column 2
        qui reg `tmpLHS' `tmpRHS2'
 10.          matrix chkCL[count,2] = e(r2)
 11.
}

// Report results:
// Simple pair-wise correlation:
corr income age age_income agesqr_income agecube_income
(obs=40)

             |   income      age age_in~e agesqr~e agecub~e
-------------+---------------------------------------------
      income |   1.0000
         age |   0.4685   1.0000
  age_income |   0.9812   0.5862   1.0000
agesqr_inc~e |   0.9436   0.6504   0.9893   1.0000
agecube_in~e |   0.8975   0.6887   0.9636   0.9921   1.0000
```

```
394
395 // R^2 of Auxillary Regressions
396 matrix list chkCL

  chkCL[5,2]
                     ptC         ptF
       income  .99796285  .99982918
          age  .68399813  .82598027
   age_income  .99956346  .99998628
  agesqr_inc~e  .99858645   .9999902
  agecube_in~e  .99993763  .99993763

397
398 /* Discussion:
  >
  > There is a strong correlation among income and the various (AGE^n x INCOME)
  > interaction terms. This appears in both the pair-wise correlations, and is very
  > strong in the auxillary regressions with R2 upwards of .999 for some variables.
  > All this suggests that collinearity may be a problem.
  >
  > Below we re-run OLS for the regressions in Part (A) and Part (C) to see how the
  > estimated values change. Adding the agesqr_income term slightly increases the
  > R^2 (since the LHS variable didn't change, this is all a drop in SSE). However,
  > the adjusted R^2 falls slightly and the estimate of RMSE increases slightly,
  > showing that the increase in R^2 is not large relative to the loss of degrees
  > of freedom. The estimate of standard errors for all the variables increases,
  > showing that the effective marginal variation in each of the RHS variables
  > is declining. The jump in the standard errors is particularly large for income
  > and age_income, the variables with a large correlation with agesqr_income.
  >
  > */
399
400 reg pizza age income age_income

      Source |       SS           df       MS      Number of obs   =        40
-------------+----------------------------------   F(3, 36)        =      7.59
       Model |  367043.248         3  122347.749   Prob > F        =    0.0005
    Residual |  580608.652        36  16128.0181   R-squared       =    0.3873
-------------+----------------------------------   Adj R-squared   =    0.3363
       Total |    947651.9        39  24298.7667   Root MSE        =       127

------------------------------------------------------------------------------
       pizza |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  -2.977423   3.352101    -0.89   0.380    -9.775799    3.820952
      income |   6.979905   2.822768     2.47   0.018     1.255067    12.70474
  age_income |  -.1232393   .0667187    -1.85   0.073    -.2585512    .0120725
       _cons |   161.4654   120.6634     1.34   0.189     -83.2513    406.1822
------------------------------------------------------------------------------

401
402 reg pizza age income age_income agesqr_income

      Source |       SS           df       MS      Number of obs   =        40
-------------+----------------------------------   F(4, 35)        =      5.83
       Model |  378782.696         4  94695.6741   Prob > F        =    0.0011
    Residual |  568869.204        35  16253.4058   R-squared       =    0.3997
-------------+----------------------------------   Adj R-squared   =    0.3311
       Total |    947651.9        39  24298.7667   Root MSE        =    127.49

-------------------------------------------------------------------------------
        pizza |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
--------------+----------------------------------------------------------------
          age |  -2.038273   3.541904    -0.58   0.569     -9.22872    5.152173
       income |   14.09616   8.839862     1.59   0.120    -3.849713    32.04203
   age_income |  -.4703705   .4139079    -1.14   0.264    -1.310648    .3699071
agesqr_income |   .0042048   .0049476     0.85   0.401    -.0058393    .0142488
        _cons |   109.7208   135.5725     0.81   0.424     -165.506    384.9475
-------------------------------------------------------------------------------
```

```
//Convert log file (smcl) to pdf
```