Statistics/Data Analysis

```
      name:  <unnamed>
       log:  C:\Users\Conor\Documents\Conor\Grad School\TA Work\Econ 103 - Econometric
> s\STATA Work\Week 3\wk3_section_log.smcl
  log type:  smcl
 opened on:  22 Jan 2018, 15:16:07

. 
. // Demonstration STATA code for week 3
. // Principles of Econometrics 4th Edition
. // Covered Problems: 3.6, 4.13
. 
. set more off

. clear all

. 
. // Create a sub-directory to store figure output into
. capture mkdir "./Figures"

. 
. ////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> ////////////////////////////////// Question 3.6 \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> ////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> 
. ********************************************************************************
. *Setup: We consider data on a motel that underwent repairs to fix defects in
. * some of the rooms. It took seven months to correct the defects, during which
. * approximately 14 rooms in the 100-unit motel were taken out of service for one
. * month at a time. The data are in motel.dta
. *
. * Parts (A) - (F)
. ********************************************************************************
. 
. use motel.dta, clear

. 
. ********************************************************************************
. *3.6 Part A: In the linear regression model MOTEL_PCT = beta1 + beta2*COMP_PCT + e,
. * test the null hypothesis H0: beta2 <= 0 against the alternative hypothesis
. * H1: beta2 > 0 at alpha = 0.01 level of significance. Discuss your conclusion.
. * Include in your answer a sketch of the rejection region and a calculation of
. * the p-value.
. ********************************************************************************
. 
. //To double check the meaning of the variables, we can use the "describe" command
. // (desc for short) to have STATA report the variable label.
. desc motel_pct comp_pct

              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------------
motel_pct       double  %10.0g                percentage motel occupancy
comp_pct        double  %10.0g                percentage competitors occupancy

. reg motel_pct comp_pct

      Source |       SS           df       MS      Number of obs   =        25
-------------+----------------------------------   F(1, 23)        =     18.19
       Model |  2208.92033         1  2208.92033   Prob > F        =    0.0003
    Residual |  2792.52127        23  121.413968   R-squared       =    0.4417
-------------+----------------------------------   Adj R-squared   =    0.4174
       Total |   5001.4416        24    208.3934   Root MSE        =    11.019

------------------------------------------------------------------------------
   motel_pct |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    comp_pct |   .8646393   .2027119     4.27   0.000     .4452978    1.283981
       _cons |   21.39999   12.90686     1.66   0.111    -5.299896    48.09987
------------------------------------------------------------------------------
```

```
//Since our null is for beta2 = 0, the t-stat reported by STATA matches the one
// we need for our test. We next need to calculate the appropriate critical value
// --> Because we have very few observations (25), the t-distribution will
//     have wider tails, so we will have a larger critical value than normal
scalar criticalT_01_1side = invttail(e(df_r),0.01)

disp "Alpha = 0.01 Critical T-Value for RHS rejection region, DF = " e(df_r) ": " cr
> iticalT_01_1side
Alpha = 0.01 Critical T-Value for RHS rejection region, DF = 23: 2.4998667


// Even with this large t critical value, we are able to reject the null of
// beta2 = 0 in favor of beta2 > 0

// Alternatively, to conduct our test we could calculate a p value. This would
// be given as follows:
scalar pval = ttail(e(df_r),_b[comp_pct]/_se[comp_pct])

disp "P-Value for H0: beta2 <= 0 vs. H1: beta2 > 0, DF = " e(df_r) ": " pval
P-Value for H0: beta2 <= 0 vs. H1: beta2 > 0, DF = 23: .00014531


//Since pval is 0.00014531, for any confidence level above that number (for
// example, 0.005 or 0.001) we would still reject the null that beta2 <= 0 in
// favor of the alternative beta2 > 0

/*Discussion:
> We can imagine two extreme cases for how competition between the
> two motels works:
> (1) There is a fixed number of visitors each period and
>         the motels compete to snag more business (e.g. there is usually 1 wedding
>         or conference per week and all the customers go to 1 or the other). In this
>         case, we would expect occupancy rates to be negatively correlated.
>
> (2) The town overall has variation in the number of customers and they
>         (roughly equally) go to each motel. For example, there is a tourist season
>         when all the motels are full and a slow season when the motels are mostly
>         empty. In economics language, we would say that the motel and its competitor
>         face the same demand shocks. In this case, we would expect occupancy rates
>         to be positively correlated.
>
> Our finding here, that the competitor and our own occupancy rates are highly
> positively correlated (point estimate of about 0.88) lends support to our
> scenario (2).
> */

////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> ///////////////////////////// Preparing a Figure \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
>
// Create the requested figure:
// (1) store the degrees of freedom from the last regression
// (2) clear the dataset and tell stata to create a blank workspace with 500 observa
> tions
// (3) use STATA to generate a row of data 1 to 500 by using gen tcdf = _n
// (4) Convert 1 to 500 to a range 1/501 to 500/501, separated by 500 steps
//     --> note that invt(dfree, 0) and invt(dfree, 1) don't make sense since
//     --> the t-distribution can take on values from -infty to +infty
// (5) Use the invt function to convert probabilites tcdf to t-values (tval)
// (6) Use ntden to convert t values to values from PDF of a t (tpdf)
```

```
// (7) Graph the line of the pdf
// (8) Graph the area above the critical value we calculated earlier
scalar dfree = e(df_r)

clear

set obs 500
 number of observations (_N) was 0, now 500

gen tcdf = _n

replace tcdf = _n/(_N+1) //_n = row of data, while _N = total # or rows
 (500 real changes made)

gen tval = invt(dfree,tcdf)

gen tpdf = ntden(dfree,0,tval)

scalar criticalT_01_2side = invttail(dfree,0.01/2)

twoway (line tpdf tval) ///
 >                (area tpdf tval if tval>criticalT_01_1side, legend(label(2 "Rejectio
 > n Region (1-sided)")) color(red)) ///
 >                (area tpdf tval if tval>criticalT_01_2side, legend(label(3 "Rejectio
 > n Region (2-sided)")) fint(inten20) color(green)) ///
 >                (area tpdf tval if tval<(-1)*criticalT_01_2side, fint(inten20) color
 > (green)), ///
 >                        ytitle("T-Distribution PDF - f(x)") xtitle("T-stat Value") /
 > //
 >                        title("Q 3.6A: T-test with Right-Side Rejection Region") ///
 >                        legend(order(2 3))

graph export "./Figures/Q 3-6A Right Side Test.pdf", replace
 (file ./Figures/Q 3-6A Right Side Test.pdf written in PDF format)


// Note that the figure is identical if we instead shaded the region tcdf>0.99
// twoway (line tpdf tval) (area tpdf tval if tcdf> 1-0.01)

// Remove the data for the figure and bring back the motel data
use motel.dta, clear


///////////////////////////// End Figure Preparation \\\\\\\\\\\\\\\\\\\\\\\\\\\\\
 > ////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
 >
*****************************************************************************
*3.6 Part B: Consider a linear regression with y = MOTEL_PCT and x = RELPRICE,
* which is the ratio of the price per room charged by the motel in question relative
* to its competitors. Test the null hypothesis that there is no relationship
* between these variables against the alternative that there is an inverse
* relationship between them at the alpha = 0.01 level of significance. Discuss
* your conclusion. Include in your answer a sketch of the rejection region, and
* a calculation of the p-value. In this exercise follow and SHOW all the test
* procedure steps suggested in Chapter 3.4
*****************************************************************************

// Test Procedures
// (1) Determine Null and Alternative Hypotheses
//     --> H0: beta2>=0   H1: beta2<0
```

```
// (2) Specify test statistic and its distribution under the null
//     --> Test Statistic: t = b2/se(b2)   Distribution, T(n-2) in this case 25
// (3) Select alpha and determine rejection region
//     --> alpha = 0.01   rejection region, t < -criticalT_01_1side
// (4) Calculate sample value of test statistic (see regression output)
// (5) State your conclusion (see below)

reg motel_pct relprice

      Source |       SS           df       MS      Number of obs   =        25
-------------+----------------------------------   F(1, 23)        =      4.38
       Model |  800.090527         1  800.090527   Prob > F        =    0.0476
    Residual |  4201.35107        23  182.667438   R-squared       =    0.1600
-------------+----------------------------------   Adj R-squared   =    0.1234
       Total |   5001.4416        24    208.3934   Root MSE        =    13.515

------------------------------------------------------------------------------
   motel_pct |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    relprice |  -122.1186   58.35027    -2.09   0.048    -242.8253   -1.411883
       _cons |    166.656   43.57095     3.82   0.001     76.52262    256.7894
------------------------------------------------------------------------------

disp "Alpha = 0.01 Critical T-Value for LHS rejection region, DF = " dfree ": " (-1)
> *criticalT_01_1side
Alpha = 0.01 Critical T-Value for LHS rejection region, DF = 23: -2.4998667


// The t-statistic is -2.09 versus the critical value of -2.50 so we fail to
// reject the null of beta2 >= 0 at the 0.01 significance level.

scalar pval = 1-ttail(e(df_r),_b[relprice]/_se[relprice])

disp "P-Value for H0: beta2 >= 0 vs. H1: beta2 < 0, DF = " e(df_r) ": " pval
P-Value for H0: beta2 >= 0 vs. H1: beta2 < 0, DF = 23: .02379463


/* Discussion:
> One of the main ways that the motels might try to compete against each other is
> by adjusting their relative prices. In general, we would expect that a cut in
> relative price should bring in more customers. This would correspond to the
> beta estimate in the above regression being negative. The point estimate agrees
> with this idea: interpreting it directly, relprice is in decimal units, it says
> that a 1 p.p. decline (i.e. -0.01) in relative price leads to a 1.2 p.p. increase
> in occupancy rate. However, the standard error on the estimate is very large, at
> almost half the magnitude of the point estimate. This, together with the low
> sample size (which drives up the tails of the t-distribution), makes it hard to
> reject the null that there is no effect or a positive effect on occupancy from
> changes in relative price.
>
> Comparing this result to what we saw in part (a), the main driver of the
> difference in the t-statistic is likely that there isn't large variation
> in the relative price. Recall that
> var(b2) = sigma_hat^2 / (sum (xi-xbar)^2) = sigma_hat^2 / ( (n-1) * se(x)^2)
> or put another way, we have
> se(b2) = sigma_hat / (sqrt(n-1)*se(x))
>
> Relative to the regression with comp_pct, the sigma_hat (Root MSE in stata output)
> is a bit larger (11 vs 13.5) the standard error of relprice (0.047, or 4.7 if we
> scale decimals up to percentage points by using 100*relprice) is much smaller
> than the standard error of comp_pct (11.1). It is easy to check these values by
> using sum relprice comp_pct
> */
```

```

//////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
  > ////////////////////////////// Preparing a Figure \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
  >
// To generate the requested figure, use the same steps as discussed above,
// except this time we have a left-hand-side rejection region
// --> Note that we already have the scalar variable dfree stored, so we don't
//     have to generate it again
clear

set obs 500
  number of observations (_N) was 0, now 500

gen tcdf = _n

replace tcdf = _n/_N //_n = row of data, while _N = total # or rows
  (500 real changes made)

gen tval = invt(dfree,tcdf)
  (1 missing value generated)

gen tpdf = ntden(dfree,0,tval)
  (1 missing value generated)

twoway (line tpdf tval) ///
  >                (area tpdf tval if tval<(-1)*criticalT_01_1side, legend(label(2 "Rej
  > ection Region (1-sided)")) color(red)) ///
  >                (area tpdf tval if tval>criticalT_01_2side, legend(label(3 "Rejectio
  > n Region (2-sided)")) fint(inten20) color(green)) ///
  >                (area tpdf tval if tval<(-1)*criticalT_01_2side, fint(inten20) color
  > (green)), ///
  >                        ytitle("T-Distribution PDF - f(x)") xtitle("T-stat Value") /
  > //
  >                        title("Q 3.6B: T-Test with Left-Side Rejection Region") ///
  >                        legend(order(2 3))


graph export "./Figures/Q 3-6B Left Side Test.pdf", replace
  (file ./Figures/Q 3-6B Left Side Test.pdf written in PDF format)


// Note that the figure is identical if we instead shaded the region tcdf<0.01
// twoway (line tpdf tval) (area tpdf tval if tcdf< 0.01)

// Remove the data for the figure and bring back the motel data
use motel.dta, clear


////////////////////////////// End Figure Preparation \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
  > //////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
  >

********************************************************************************
*3.6 Part C: Consider the linear regression MOTEL_PCT = delta1 + delta2*REPAIR + e
* where REPAIR is an indicator variable taking the value 1 during the repair
* period and 0 otherwise. Test the null hypothesis H0: delta2 >= 0 against the
* alternative hypothesis H1: delta2 < 0 at the alpha = 0.05 level of
* significance. Explain the logic behind stating the null and alternative
* hypotheses in this way. Discuss your conclusions.
********************************************************************************
```

```
//Note the interpretation of the indicator variable:
//--> When using only an indicator variable on the right hand side, the constant
//    term is the average of the left-hand side term WHEN THE INDICATOR VARIALBES
//    ARE ALL ZERO (aka the average for the excluded group). Then the beta is
//    equal to the DIFFERENCE IN THE AVERAGE between the excluded group (IND = 0)
//    and the specified group (IND = 1).
//
//For example, in the regression results below, the b1 estimate is equal to the
// average of motel_pct when repair == 0 while b1 + b2 is equal to the average
// of motel_pct when repair==1

reg motel_pct repair
```

Source	SS	df	MS
Model	**882.928029**	**1**	**882.928029**
Residual	**4118.51357**	**23**	**179.065807**
Total	**5001.4416**	**24**	**208.3934**

Number of obs	=	**25**
F(1, 23)	=	**4.93**
Prob > F	=	**0.0365**
R-squared	=	**0.1765**
Adj R-squared	=	**0.1407**
Root MSE	=	**13.382**

motel_pct	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
repair	**-13.23571**	**5.960615**	**-2.22**	**0.037**	**-25.56619**	**-.9052429**
_cons	**79.35**	**3.154061**	**25.16**	**0.000**	**72.82533**	**85.87467**

```
sum motel_pct if repair == 0
```

Variable	Obs	Mean	Std. Dev.	Min	Max
motel_pct	**18**	**79.35**	**11.64592**	**62.9**	**96.2**

```
lincom repair + _cons
```

(1) **repair + _cons = 0**

motel_pct	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	**66.11429**	**5.057749**	**13.07**	**0.000**	**55.65153**	**76.57704**

```
sum motel_pct if repair == 1
```

Variable	Obs	Mean	Std. Dev.	Min	Max
motel_pct	**7**	**66.11429**	**17.38222**	**39.2**	**82.4**

```

// With the interpretation of the cofficients for an indicator regression in
// mind, we can see that we have two ways of asking the same question:
// (1) Was the occupancy rate lower, on average, during the 7 months of the
//     repair period, compared to the other 18 months in our sample?
// (2) Is b2 < 0?
//
// Given that the occupancy rate fluctuates from one month to the next, we would
// need to use a t-test to see if any observed decline during the repair period
// is extreme enough that we couldn't reasonably blame any decline on the typical
```

```
// variance in occupancy rates.
//
// Once again, the t-stat STATA reported is the one we need, so all that's left is
// to compare that t-stat value to the appropriate critical value. However, we
// don't even need to do that, since STATA already reported the 95% confidence
// interval. Given that 0 is outside the confidence interval, we know we will
// reject the null at the same confidence level for a 1-sided test.

scalar criticalT_05_1side = invttail(e(df_r),0.05)

disp "Alpha = 0.05 Critical Value for t-test w/ LHS rejection region, DF = " e(df_r)
>  ": " (-1)*criticalT_05_1side
Alpha = 0.05 Critical Value for t-test w/ LHS rejection region, DF = 23: -1.7138715

disp "T-stat for H0: delta2 (>)= 0: " _b[repair]/_se[repair]
T-stat for H0: delta2 (>)= 0: -2.2205283


*******************************************************************************
*3.6 Part D: Using the model given in part (c), construct a 95% interval estimate
* for the parameter delta2 and give its interpretation. Have we estimated the
* effect of the repairs on motel occupancy relatively precisely, or not? Explain.
*******************************************************************************

// STATA already reported the 95% confidence interval, but here I review how to
// calculate it by hand
scalar criticalT_05_2side = invttail(e(df_r),0.05/2)

scalar ciLow_repair = _b[repair]-criticalT_05_2side*_se[repair]

scalar ciHigh_repair = _b[repair]+criticalT_05_2side*_se[repair]

disp "95% Confidence Interval: [" ciLow_repair ", " ciHigh_repair "]"
95% Confidence Interval: [-25.566186, -.9052429]


// The interpretation for the confidence interval is that, in a 2-sided test at
// the 95% confidence level, we cannot reject the null that delta2 = c for any
// value of c between ciLow (-25.6) and ciHigh (-0.9)

/* Discussion:
> The confidence interval is quite wide, with the effect varying from less than
> a 1 percentage point drop to over a 25 percentage point drop. A 25 percentage
> point effect would be about 3/4 of the gap between the highest occupancy rate
> and the lowest occupancy rate in the non-repair period. This wide estimation
> band reflects the small sample size and the substantial variation in occupancy
> rates during both the repair and non-repair periods.
> */

*******************************************************************************
*3.6 Part E: Consider the linear regression model with y = MOTEL_PCT - COMP_PCT
* and x = REPAIR that is (MOTEL_PCT - COMP_PCT) = gamma1 + gamma2*REPAIR + e.
* Test the null hypothesis that gamma2 = 0 against the alternative that
* gamma2 < 0 at the alpha = 0.01 level of significance. Discuss the meaning of
* the test outcome.
*******************************************************************************

// Since the degrees of freedom haven't changed, we can use the same critical
// value we calculated earlier. Again, we can compare our critical value to the
// t-statistic reported by STATA in the regression output
```

```
gen pct_diff = motel_pct - comp_pct

reg pct_diff repair
```

Source	SS	df	MS
Model	1004.59849	1	1004.59849
Residual	1842.05986	23	80.0895593
Total	2846.65835	24	118.610765

Number of obs	=	25
F(1, 23)	=	12.54
Prob > F	=	0.0017
R-squared	=	0.3529
Adj R-squared	=	0.3248
Root MSE	=	8.9493

pct_diff	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
repair	-14.11825	3.986325	-3.54	0.002	-22.3646	-5.871913
_cons	16.86111	2.109365	7.99	0.000	12.49756	21.22466

```
disp "Alpha = 0.01 Critical Value for LHS rejection region t-test, DF = " e(df_r) ":
>  " (-1)*criticalT_01_1side
```

Alpha = 0.01 Critical Value for LHS rejection region t-test, DF = 23: -2.4998667

```
// Since the new t stat is -3.54 compared to the critical value of -2.5 this
// time we reject the null that there was no (or positive) effect of the repair
// in favor of the alternaive that the repair had a negative effect.

// Let's once again compare the beta coefficient on the indicator variable
// REPAIR to the difference in the average pct_diff when repair == 0 and
// repair == 1. We will see that the beta estimate is equal to the difference in
// the means. We also compare the standard error of model_pct to that of pct_diff.
sum pct_diff motel_pct
```

Variable	Obs	Mean	Std. Dev.	Min	Max
pct_diff	25	12.908	10.89086	-9	40.8
motel_pct	25	75.644	14.43584	39.2	96.2

```
bysort repair: sum pct_diff motel_pct
```

-> repair = 0

Variable	Obs	Mean	Std. Dev.	Min	Max
pct_diff	18	16.86111	9.238802	6.6	40.8
motel_pct	18	79.35	11.64592	62.9	96.2

-> repair = 1

Variable	Obs	Mean	Std. Dev.	Min	Max
pct_diff	7	2.742857	8.072764	-9	14.8
motel_pct	7	66.11429	17.38222	39.2	82.4

```
/* Discussion:
> Overall, we find a small increase in the magnitude of the point estimate
> (-13.2 to -14.1) but a big increase in the magnitude of the t statistic
> (-2.2 to -3.5). Since the right-hand side variable (repair) is the same in both
> cases, we know there was no change in the variance of our right-hand side
> variables. Instead, most of the decline is driven by a smaller estimate of
> sigma_hat. That, in turn, mechanically reflects the smaller variance of pct_diff
> (conditional on repair) compared to the variance in motel_pct.
>
> So did we cheat by switching motel_pct to pct_diff, especially if what we really
> care about is the effect on motel_pct? It depends on your stance about what
> variation in motel_pct we should and shouldn't be paying attention to.
>
```

```
> I would argue that it makes sense to make the switch to pct_diff. One criticism
> of the regression in Part (D) is that mechanically all we're calculating is the
> change in average occupancy between the 7 months of the repair period and all
> other times, so other random stuff that happened during that period could have
> caused a drop in occupancy in addition to the repairs. However, from our
> regression in Part A, we know that the occupany rate in the competitor (comp_pct)
> is a decent predictor of occupancy at our own motel. One interpretation of what
> we're doing is we are partly controlling for (unobserved) demand shocks by using
> the competitor occupancy as a proxy. These demand shocks (for example, if repairs
> were made during the low season in the winter) would add variance to motel_pct
> that hides the true effect, so we would want to try to get rid of this type of
> noise/variance in the data.
>
> Let's phrase this in terms of an ad hoc economic model. Imagine that the truth
> is that motel_pct and comp_pct behave as follows:
>
>         motel_pct = alpha1 + alpha2*repair + (d + a)
>         comp_pct  = delta1 + (d + b)
>
> where d, a, and b are all unobserved shocks which are normal, mean zero, have
> constant variance and are all mutually independent. Using economic language, we
> can say that d is a common demand shock, while a and b are idiosyncratic shocks
> for the motel and the competitor, respectively.
>
> Below, I will use (#) to refer to the "true" economic model in terms of the two
> equations shown above, and (#a) to refer to the "econometric" or "reduced form"
> model that is being fed into the OLS procedure.
>
> Given the economic model I proposed, consider two options for estimating alpha2:
>
>         (1)  motel_pct = alpha1 + alpha2*repair + (d + a)
>         (1a) motel_pct = beta1 + beta2*repair + e(1)
>
>         (2)  (motel_pct - comp_pct) = (alpha1 - delta1) + alpha2*repair + (a-b)
>         (2a) (motel_pct - comp_pct) = gamma1 + gamma2*repair + e(2)
>
> Given our assumptions about d, a, and b both options (1a) and (2a) satisfy all
> the OLS assumptions, so both are valid regressions. While the meaning of the
> constant in the regression changes (beta1 estimates alpha1 while gamma1 estimates
> alpha1 - delta1), the term on repair is an estimator for alpha2 in both cases.
> Which estimator should we prefer, beta2 or gamma2? Well, we know the variance of
> beta2 depends on the variance of e(1) while the variance of gamma2 depends on
> the variance of e(2). Given our assumptions, Var(e(1)) = Var(d) + Var(a) and
> Var(e(2)) = Var(a) + Var(b). So the question is whether we think the variance of
> the common shocks d is larger or smaller than the idiosyncratic shocks b. Given
> our results, it seems likely that Var(d) > Var(b).
>
> We can also use this model to rule out an alternative regression that might
> seem to have an intuitive appeal. Earlier we said comp_pct is acting like a
> control for the unobserved common demand shock, so would it be reasonble to try
> running OLS as follows:
>
>         (3a) motel_pct = z1 + z2*repair + z3*comp_pct + e(3)
>
> Mapping this regression model back to our simple economic model, the economic
> model would say that the truth is:
>
>         (3)  motel_pct = (alpha1-delta1) + alpha2*repair + (1)*(delta1+d+b) + (a-b)
>
> We would want to say that z1 is an estimator for (alpha1 - delta1), z2 is
> an estimator for alpha2 and z3 is an estimator for the number 1. However, our
> economic model tells us that this regression violates the OLS assumptions. Notice
> that the shock for the competitor (b) appears in both the right-hand side variable
> comp_pct and in the error term e(3). The bias introduced here could be thought of
> as measurement error for the common shock d, which results in "attenuation bias"
> reflected in an estimate for z3 that is below the magnitude of the true value.
> This attenuation bias also potentially affects the estimates for z1 and z2.
>
> To (potentially) see attenuation bias in action, let's think about the regression
> from Part A in terms of the economic model I proposed:
>
>         (4a) motel_pct = x1 + x2*comp_pct + e(4)
```

```
  >        (4)  motel_pct=(alpha1-delta1+avg(alpha2*repair)) + (1)*(delta1+d+b) + (a-b)
  >
  > Notice that x2 is supposed to be an estimator for the number (1), but that we
  > have this issue that (b) appears in both comp_pct and the error term e(4). In
  > Part A we found a point estimate for x2 that was about 0.86 and failed to reject
  > the null that x2 >= 1. However, if we had a large sample the attenuation bias
  > would remain and we would eventually be able to reject the null that x2 >=1.
  > */
236
237 ////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
  > ///////////////////////////// Preparing a Figure \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
  >
238 //To fix ideas, let's plot the data and fitted values for part (d) and part (e)
239 //--> Note: Don't get too worried about where I put in missing value, this is just
240 //          to help make the figure look nice.
241 sort time

242 reg motel_pct repair
```

Source	SS	df	MS
Model	882.928029	1	882.928029
Residual	4118.51357	23	179.065807
Total	5001.4416	24	208.3934

Number of obs	=	25
F(1, 23)	=	4.93
Prob > F	=	0.0365
R-squared	=	0.1765
Adj R-squared	=	0.1407
Root MSE	=	13.382

motel_pct	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
repair	-13.23571	5.960615	-2.22	0.037	-25.56619	-.9052429
_cons	79.35	3.154061	25.16	0.000	72.82533	85.87467

```
243 predict fit_partD, xb

244 gen fit_partD_norepair = fit_partD

245 replace fit_partD_norepair = . if repair == 1 // Put in "missing value" when repair
  > == 1
  (7 real changes made, 7 to missing)

246 gen fit_partD_repair = fit_partD

247 replace fit_partD_repair = . if repair == 0 // Put in "missing value" when repair ==
  >  0
  (18 real changes made, 18 to missing)

248 twoway (line fit_partD_norepair time, cmissing(n) lcolor(blue) legend(label(1 "Fitte
  > d: Repair == 0"))) ///
  >                 (line fit_partD_repair time, cmissing(n) lcolor(red) legend(label(2
  > "Fitted: Repair == 1"))) ///
  >                 (scatter motel_pct time if repair == 0, mcolor(blue) legend(label(3
  > "Data: Repair == 0"))) ///
  >                 (scatter motel_pct time if repair == 1, mcolor(red) legend(label(4 "
  > Data: Repair == 1"))), ///
  >                         ytitle("Motel Occupancy Rate (%)") xtitle("Time") ///
  >                         title("Motel Occupancy") subtitle("Repair and Non-Repair Per
  > iods") ///
  >                         text(45 1 "Model: motel_pct = b1 + b2*repair", place(e))
```

```
249
250 graph export "./Figures/Q 3-6 Motel_Pct Regression.pdf", replace
  (file ./Figures/Q 3-6 Motel_Pct Regression.pdf written in PDF format)

251
252 reg pct_diff repair

      Source |       SS           df       MS      Number of obs   =        25
-------------+----------------------------------   F(1, 23)        =     12.54
       Model |  1004.59849         1  1004.59849   Prob > F        =    0.0017
    Residual |  1842.05986        23  80.0895593   R-squared       =    0.3529
-------------+----------------------------------   Adj R-squared   =    0.3248
       Total |  2846.65835        24  118.610765   Root MSE        =    8.9493

------------------------------------------------------------------------------
    pct_diff |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      repair |  -14.11825   3.986325    -3.54   0.002     -22.3646   -5.871913
       _cons |   16.86111   2.109365     7.99   0.000     12.49756    21.22466
------------------------------------------------------------------------------

253 predict fit_partE, xb

254 gen fit_partE_norepair = fit_partE

255 replace fit_partE_norepair = . if repair == 1 // Put in "missing value" when repair
  > == 1
  (7 real changes made, 7 to missing)

256 gen fit_partE_repair = fit_partE

257 replace fit_partE_repair = . if repair == 0 // Put in "missing value" when repair ==
  >  0
  (18 real changes made, 18 to missing)

258 twoway (line fit_partE_norepair time, cmissing(n) lcolor(blue) legend(label(1 "Fitte
  > d: Repair == 0"))) ///
  >                 (line fit_partE_repair time, cmissing(n) lcolor(red) legend(label(2
  > "Fitted: Repair == 1"))) ///
  >                 (scatter pct_diff time if repair == 0, mcolor(blue) legend(label(3 "
  > Data: Repair == 0"))) ///
  >                 (scatter pct_diff time if repair == 1, mcolor(red) legend(label(4 "D
  > ata: Repair == 1"))), ///
  >                         ytitle("Motel - Competitor Occupancy Rate (p.p.)") xtitle("T
  > ime") ///
  >                         title("Gap in Occupany Rate vs Competitor") subtitle("Repair
  >  and Non-Repair Periods") ///
  >                         text(-5 1 "Model: pct_diff = b1 + b2*repair", place(e))

259
260 graph export "./Figures/Q 3-6 Pct_Diff Regression.pdf", replace
  (file ./Figures/Q 3-6 Pct_Diff Regression.pdf written in PDF format)

261
262 ///////////////////////////// End Figure Preparation \\\\\\\\\\\\\\\\\\\\\\\\\\
  > ////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
  >
263 ********************************************************************************
264 *3.6 Part F: Using the model in part (e), construct and discuss the 95% interval
265 * estimate of gamma2.
```

```
********************************************************************************

// STATA already calculated the 95% confidence interval above in the output
// for reg pct_diff repair

/* Discussion:
> As we noted earlier, the magnitude of the t statistic is larger in part (e)
> because of a larger point estimate and a smaller se(b2). The same forces are
> moving the confidence interval around. First, the confidence interval is shifted
> lower because of the lower (more negative) point estimate. Second, the width
> of the confidence interval shrank because of the smaller se(b2). Both forces
> tend to bring ciHigh down, but they work in opposite direction for ciLow; we
> can see that most of the impact is coming from se(b2) because ciLow is higher
> (less negative) than it was in part (d) despite the lower point estimate.
> */

////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> ///////////////////////////////// Question 4.13 \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> ////////////////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
>
********************************************************************************
*Setup: Consider data on 880 houses sold in Stockton, CA during mid-2005
*
* Parts (A) - (H)
********************************************************************************

use stockton2.dta, clear


********************************************************************************
*4.13 Part A: Estimate the log-linear model in ln(PRICE) = beta1 + beta2*SQFT + e.
* Interpret the estimated model parameters. Calculate the slope and elasticity
* at the sample means, if necessary.
********************************************************************************

gen ln_price = log(price)

reg ln_price sqft

      Source |       SS           df       MS      Number of obs   =       880
-------------+----------------------------------   F(1, 878)       =   2143.38
       Model |  88.3556977         1  88.3556977   Prob > F        =    0.0000
    Residual |  36.1934444       878  .041222602   R-squared       =    0.7094
-------------+----------------------------------   Adj R-squared   =    0.7091
       Total |  124.549142       879  .141694132   Root MSE        =    .20303

------------------------------------------------------------------------------
    ln_price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        sqft |    .000596   .0000129    46.30   0.000     .0005707    .0006212
       _cons |   10.59379     .02185   484.84   0.000      10.5509    10.63667
------------------------------------------------------------------------------


// STATA Note: to have STATA run a command but NOT report the output in the
// results window, you can put quietly (or qui for short) before a command.
// We don't want to see all the summary detail for sqft and price, we just want
// to use STATA to calculate the mean for later use, so we use qui here
qui sum sqft
```

```
scalar mean_sqft = r(mean)

qui sum price

scalar mean_price = r(mean)

// Slope: since y = exp(sigma^2/2)*exp(b1 + b2*x), we have
// --> dy/dx = exp(sigma^2/2)*exp(b1+b2*x)*b2 = y*b2
lincom `=mean_price'*_b[sqft]

 ( 1)  112810.8*sqft = 0

------------------------------------------------------------------------------
    ln_price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   67.23106    1.45218    46.30   0.000     64.38091     70.0812
------------------------------------------------------------------------------

// Elasticity: Given our model, dlny/dlnx = (x/y)*(dy/dx) = x*b2
lincom `=mean_sqft'*_b[sqft]

 ( 1)  1611.968*sqft = 0

------------------------------------------------------------------------------
    ln_price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   .9606732   .0207504    46.30   0.000     .9199471    1.001399
------------------------------------------------------------------------------


// Compare estimate of the elasticity using lincom to the margins command using
// the dyex option. The dyex tells stata to calculate the "margin" of the form
// dy/dlnx, but since dy is already dlny, and the elasticity is equal to
// dlny/dlnx this gets us to our desired answer
margins, dyex(sqft) atmeans

Conditional marginal effects                    Number of obs     =        880
Model VCE    : OLS

Expression   : Linear prediction, predict()
dy/ex w.r.t. : sqft
at           : sqft            =    1611.968 (mean)

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/ex   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        sqft |   .9606732   .0207504    46.30   0.000      .919947    1.001399
------------------------------------------------------------------------------


// Store residuals and fitted values for later
predict ln_price_hat_loglin, xb

predict residual_loglin, residual

label var residual_loglin "Residual (Log-Linear)"

gen price_hat_loglin = exp(ln_price_hat_loglin)
```

```
gen price_hat_loglin_adju = price_hat_loglin*exp(e(rmse)^2/2)

//stdf option for predict = standard error of forecast
//aka s.e.(yi - yhati) where yi refers to OLS regression (not any transformations)
predict stdf_loglin, stdf


*****************************************************************************
*4.13 Part B: Estimate the log-log model ln(PRICE) = beta1 + beta2*ln(SQFT) + e.
* Interpret the estimated parameters. Calculate the slope and elasticity at the
* sample means, if necessary.
*****************************************************************************

gen ln_sqft = log(sqft)

reg ln_price ln_sqft

      Source |       SS           df       MS      Number of obs   =       880
-------------+----------------------------------   F(1, 878)       =   1993.88
       Model |  86.4716562         1  86.4716562   Prob > F        =    0.0000
    Residual |  38.0774859       878  .043368435   R-squared       =    0.6943
-------------+----------------------------------   Adj R-squared   =    0.6939
       Total |  124.549142       879  .141694132   Root MSE        =    .20825

------------------------------------------------------------------------------
    ln_price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     ln_sqft |   1.006582   .0225423    44.65   0.000     .9623386    1.050825
       _cons |   4.170677   .1655084    25.20   0.000     3.845839    4.495515
------------------------------------------------------------------------------


// In a log-log model, the beta coefficent corresponds to an elasticity, so we
// don't need to do any more calculation for that.
// The formula for slope dy/dx can be found by noting that
// y = exp(b1+b2*ln(x)) = exp(b1)*x^b2
// --> dy/dx = exp(b1)*b2*x^(b2-1) = (y/x)*b2
lincom `=mean_price'/`=mean_sqft'*_b[ln_sqft]

 ( 1)  69.98327*ln_sqft = 0

------------------------------------------------------------------------------
    ln_price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   70.44388   1.577587    44.65   0.000      67.3476    73.54017
------------------------------------------------------------------------------


// Store residuals and fitted values for later
predict ln_price_hat_loglog, xb

predict residual_loglog, residual

label var residual_loglog "Residual (Log-Log)"

gen price_hat_loglog = exp(ln_price_hat_loglog)

gen price_hat_loglog_adju = price_hat_loglog*exp(e(rmse)^2/2)

//stdf option for predict = standard error of forecast
```

```
//aka s.e.(yi - yhati) where yi refers to OLS regression (not any transformations)
predict stdf_loglog, stdf

***************************************************************************
*4.13 Part C: Compare the R2 value from the linear model PRICE = beta1 + beta2*SQFT
> + e
* to the "generalized" R2 measure for the models in (b) and (c).
***************************************************************************

reg price sqft
```

Source	SS	df	MS
Model	**1.6479e+12**	**1**	**1.6479e+12**
Residual	**8.0391e+11**	**878**	**915618929**
Total	**2.4518e+12**	**879**	**2.7893e+09**

Number of obs	=	**880**
F(1, 878)	=	**1799.75**
Prob > F	=	**0.0000**
R-squared	=	**0.6721**
Adj R-squared	=	**0.6717**
Root MSE	=	**30259**

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
sqft	**81.38899**	**1.918489**	**42.42**	**0.000**	**77.62363**	**85.15435**
_cons	**-18385.65**	**3256.424**	**-5.65**	**0.000**	**-24776.94**	**-11994.37**

```
predict price_hat_lin, xb

// The "generalized" R2 measure is the square of the correlaton between the
// fitted values (yhat) and the actual values of the dependent variable (y)
//
// (1) Use STATA to calculate the correlation coefficients
corr price price_hat_lin price_hat_loglin price_hat_loglog price_hat_loglin_adju pri
> ce_hat_loglog_adju
(obs=880)
```

	price	pri~_lin	pri~glin	price_~g	p~n_adju	p~g_adju
price	**1.0000**					
price_h~_lin	**0.8198**	**1.0000**				
price_h~glin	**0.8455**	**0.9546**	**1.0000**			
price_hat_~g	**0.8201**	**1.0000**	**0.9549**	**1.0000**		
price~n_adju	**0.8455**	**0.9546**	**1.0000**	**0.9549**	**1.0000**	
price~g_adju	**0.8201**	**1.0000**	**0.9549**	**1.0000**	**0.9549**	**1.0000**

```
// --> The correlations that we care about are the values in the first column
//     in rows 2 and on. The diagonals are all equal to 1 because by definition
//     the correlation of a variable with itself must be 1.
//
// --> Note that the correlation coefficients are identical for the adjusted
//     and unadjusted versions of the log-linear and log-log models. In other
//     words, the values in rows 3 and 5 match along with rows 4 and 6. This is
//     guaranteed to be the case because the adjustment only involved multiplying
//     by a fixed number, so it ends up cancelling in the top and bottom of the
//     correlation calculation.

// Calculate "generalized" R2 by hand using numbers reported by corr
scalar r2_lin = 0.8198^2
```

```
scalar r2_loglin = 0.8455^2

scalar r2_loglog = 0.8201^2


// View Results
disp "R2 - Linear: " r2_lin
  R2 - Linear: .67207204

disp "R2 - Log-Linear: " r2_loglin
  R2 - Log-Linear: .71487025

disp "R2 - Log-Log: " r2_loglog
  R2 - Log-Log: .67256401


/////////////////////////// Random STATA Tip \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
> // We could have done the corr command above faster using the character * wildcard.
// By typing price_hat*, STATA will find ALL the workspace variables that begin
// with price_hat regardless of what follows.
// Note that in the output the order will the order of variables in the
// workspace (e.g. higher in variable list, closer to left hand side of the
// workspace browser) when you use * to fill in all the names. Try using the
// command input:
// corr price price_hat*

/////////////////// Warning: Advanced STATA Usage Below \\\\\\\\\\\\\\\\\\\\\\
> ///////////////// Feel free to ignore Steps (2) through (5) \\\\\\\\\\\\\\\\\\\\
>
// (2) The full correlation matrix is saved in the return list as r(C). Store
//     r(C) as a matrix variable.
matrix full_corr_mat = r(C)


// (3) Select only the elements of full_corr_mat that we care about: the first
//     column from rows 2, 3, and 4 (as noted above rows 5 and 6 are redundant).
matrix y_yhat_corr = full_corr_mat[2..4,1]


// (4) Calculate the square of each corr(price,price_hat), aka the square of
//     each individual element of y_yhat_corr
// Option A: Loop. Write a short program that isolates each value in
//           y_hat_corr and calculates the square of that value.
matrix r2_vectorA = [1\1\1] // create a column of ones (to be overwritten later)

forvalues x = 1/3 {
  2.         matrix r2_vectorA[`x',1] = y_yhat_corr[`x',1]^2
  3. }

// Option B: Matrix multiplication. If we take a column vector (let's call it v),
//                         then the diagonal elements of v*v' will be the square of th
> e
//           elements of v itself.
matrix r2_vectorB = vecdiag(y_yhat_corr*y_yhat_corr')'


// (5) To have STATA display the contents of a matrix, use the command input:
//     matrix list [matrix name]
matrix list r2_vectorA

r2_vectorA[3,1]
           c1
r1  .67211304
r2   .7147877
r3  .67255133
```

```
matrix list r2_vectorB

  r2_vectorB[3,1]
                      r1
  price_h~_lin  .67211304
  price_h~glin   .7147877
  price_hat_~g  .67255133

//////////////////////////// End: Advanced STATA Usage \\\\\\\\\\\\\\\\\\\\\\\\\\\
>
// Store residual and fitted values for later
predict residual_lin, residual

label var residual_lin "Residual (Linear)"

//stdf option for predict = standard error of forecast
//aka s.e.(yi - yhati) where yi refers to OLS regression (not any transformations)
predict stdf_lin, stdf

******************************************************************************
*4.13 Part D: Construct histograms of least squares residuals from each of the
* models in (a), (b), and (c) and obtain the Jarque-Bera statistics. Based on
* your observations, do you consider the distributions of the residuals to be
* compatible with an assumption of normality?
******************************************************************************

sum residual_loglin, detail

                    Residual (Log-Linear)
  -------------------------------------------------------------
        Percentiles      Smallest
   1%    -.4598814       -.710299
   5%    -.3142879      -.6619065
  10%    -.2410653      -.6477303       Obs                 880
  25%    -.1200507      -.6345798       Sum of Wgt.         880

  50%    -.0139173                      Mean          -1.73e-10
                          Largest       Std. Dev.       .202918
  75%     .1158161       .7670023
  90%     .2606355       .7681834       Variance       .0411757
  95%     .3558994       .8957195       Skewness       .3239307
  99%     .5422422       .9086631       Kurtosis       4.315611

scalar jb_loglin = (r(N)/6)*(r(skewness)^2 + ((r(kurtosis)-3)^2)/4)

sum residual_loglog, detail

                     Residual (Log-Log)
  -------------------------------------------------------------
        Percentiles      Smallest
   1%      -.49264      -.7518981
   5%    -.3091487      -.6739541
  10%    -.2441997      -.6184166       Obs                 880
  25%    -.1303633      -.5487044       Sum of Wgt.         880

  50%     -.018237                      Mean          -2.38e-10
                          Largest       Std. Dev.      .2081324
  75%     .1182303       .7180918
  90%     .2746109       .7190305       Variance       .0433191
  95%     .3657138       .8612714       Skewness       .3488042
  99%     .5302507       .8624387       Kurtosis       3.975605
```

```
scalar jb_loglog = (r(N)/6)*(r(skewness)^2 + ((r(kurtosis)-3)^2)/4)

sum residual_lin, detail

                       Residual (Linear)
-------------------------------------------------------------
      Percentiles      Smallest
 1%    -68089.01      -101224.1
 5%    -41894.24      -91337.09
10%    -30454.09      -84395.79       Obs                 880
25%    -16140.87      -76857.88       Sum of Wgt.         880

50%    -2667.711                      Mean          -9.86e-06
                        Largest       Std. Dev.      30241.98
75%     12093.73       166920.7
90%     28794.58         168413       Variance       9.15e+08
95%     50104.84       186850.3       Skewness        1.59206
99%     112023.8       204279.8       Kurtosis       10.53922

scalar jb_lin = (r(N)/6)*(r(skewness)^2 + ((r(kurtosis)-3)^2)/4)


// The distribution for the Jarque-Bera statistic is Chi Square w/ 2 degrees of
// freedom, so for each of these we can also calculate a p-value using the
// 1 minus the Chi Square (2) CDF
//
disp "JB Stat - Linear: " jb_lin ", (p-value = " 1 - chi2(2,jb_lin) ")"
  JB Stat - Linear: 2455.8747, (p-value = 0)

disp "JB Stat - Log Linear: " jb_loglin ", (p-value = " 1 - chi2(2,jb_loglin) ")"
  JB Stat - Log Linear: 78.853742, (p-value = 0)

disp "JB Stat - Log Log: " jb_loglog ", (p-value = " 1 - chi2(2,jb_loglog) ")"
  JB Stat - Log Log: 52.743634, (p-value = 3.523e-12)


hist residual_lin, kdensity title("Q 4-13: Histogram for Residuals from Linear Model
  > ")
  (bin=29, start=-101224.08, width=10534.615)

graph export "./Figures/Q 4-13 Linear Residual Histogram.pdf", replace
  (file ./Figures/Q 4-13 Linear Residual Histogram.pdf written in PDF format)


hist residual_loglin, kdensity title("Q 4-13: Histogram for Residuals from Log-Linea
  > r Model")
  (bin=29, start=-.71029896, width=.05582628)

graph export "./Figures/Q 4-13 Log-Linear Residual Histogram.pdf", replace
  (file ./Figures/Q 4-13 Log-Linear Residual Histogram.pdf written in PDF format)


hist residual_lin, kdensity title("Q 4-13: Histogram for Residuals from Log-Log Mode
  > l")
  (bin=29, start=-101224.08, width=10534.615)

graph export "./Figures/Q 4-13 Log-Log Residual Histogram.pdf", replace
  (file ./Figures/Q 4-13 Log-Log Residual Histogram.pdf written in PDF format)
```

```
/* Discussion:
> Overall, looking at the distribution and reviewing the summary statistics it is
> clear that there is a long positive tail which corresponds to the positive skew
> reported in the summary table. If the residuals were distributed normally there
> should be no skew. Kurtosis is a measure of how "fat" the tails are - i.e. how
> much of the probability mass is concentrated in events further from the mean.
> While this is harder to distinguish by looking at the histogram in isolation,
> the summary statistics show large kurtosis (greater than 3) for the residuals
> from all the models. The skew and kurtosis both contribute to large Jarque-Bera
> statistic, which leads to small p-values and clearly support rejecting the null
> that the residuals are distributed normally. This situation is most severe for
> the regressions from the linear model.
> */

********************************************************************************
*4.13 Part E: For each of the models in (a)-(c), plot the least squares residuals
* against SQFT. Do you observe any patterns?
********************************************************************************

gen zero_val = 0

twoway (scatter residual_lin sqft) (line zero_val sqft, lcolor(black)), ///
>                         title("Q 4-13: Scatter of Residuals from Linear Model") lege
> nd(off)

graph export "./Figures/Q 4-13 Linear Residual Scatter.pdf", replace
(file ./Figures/Q 4-13 Linear Residual Scatter.pdf written in PDF format)


twoway (scatter residual_loglin sqft) (line zero_val sqft, lcolor(black)), ///
>                         title("Q 4-13: Scatter of Residuals from Log-Linear Model")
> legend(off)

graph export "./Figures/Q 4-13 Log-Linear Residual Scatter.pdf", replace
(file ./Figures/Q 4-13 Log-Linear Residual Scatter.pdf written in PDF format)


twoway (scatter residual_loglog sqft) (line zero_val sqft, lcolor(black)), ///
>                 title("Q 4-13: Scatter of Residuals from Log-Log Model") legend(off)

graph export "Q 4-13 Log-Log Residual Scatter.pdf", replace
(file Q 4-13 Log-Log Residual Scatter.pdf written in PDF format)


/* Discussion:
> In all cases, the residuals appear to be more spread out (higher variance) for
> home with higher square footage. In addition, for the Linear and Log-Log models
> there is a clear tendency for the few observations in sqft>3500 to have large
> positive residuals, suggesting the model does poorly for fitting the data in
> that region.
> */

********************************************************************************
*4.13 Part F: For each of the models in (a)-(c), predict the value of a house
* with 2700 square feet.
********************************************************************************

//It turns out there are a few observations that already have sqft = 2700 so
// we can just look at the predicted values from those points
```

```
list price_hat_lin price_hat_loglin_adju price_hat_loglog_adju if sqft == 2700
```

	pri~_lin	p~n_adju	p~g_adju
556.	**201364.6**	**203515.8**	**188220.8**
668.	**201364.6**	**203515.8**	**188220.8**

```

// To avoid showing multiple observations, let's quickly find a single row number
// where sqft == 2700
gen count = _n

qui sum count if sqft == 2700

scalar first_sqft2700 = r(min)


******************************************************************************
*4.13 Part G: For each model in (a)-(c), construct a 95% prediction interval for
* the value of a house with 2700 square feet.
******************************************************************************

// For the linear model, the (1-alpha) confidence interval at each x value is
// defined as:
// --> [ yhat(x) - tc(alpha)*stdf(x), yhat(x) + tc(alpha)*stdf(x) ]
// while for regression with ln(y) on the left-hand side we have:
// --> [ exp( ln_y_hat(x) - tc(alpha)*stdf(x) ), exp( ln_y_hat(x) + tc(alpha)*stdf(x
> ) )]
// Note that tc(alpha) is the same in all circumstances
//
// Also, it's important to recall the distinction between stdf and stdp:
//       stdp = s.e.(yhat)
//       stdf = s.e.(y - yhat)
// yhat = b0 + b1*x so that the randomness in yhat comes from b0 and b1, while
// for stdf y-yhat = (beta0-b0) + (beta1-b1)+e which has randomness from b0 and
// b1 along with randomness from e. The textbook refers to the concept tied to
// stdf as the "prediction interval" and the concept for stdp as the "interval
// estimate for E(y)". Somewhat confusingly, STATA refers to stdp as the "standard
// error of the prediction", so you should be a bit careful about which concept
// is intended in which context.

// (1) Calculate critical t value for 2-sided 95% interval
// --> use regress to get degrees of freedom
qui reg price sqft

scalar criticalT_05_2side = invttail(e(df_r), 0.05/2)

// (2) Calculate low- and high- points of confidence interval for the 3 models
// --> Linear
gen cilow_price_hat_lin = price_hat_lin - criticalT_05_2side*stdf_lin

gen cihigh_price_hat_lin = price_hat_lin + criticalT_05_2side*stdf_lin

// --> Log-Linear
gen cilow_price_hat_loglin = exp(ln_price_hat_loglin - criticalT_05_2side*stdf_logli
> n)

gen cihigh_price_hat_loglin = exp(ln_price_hat_loglin + criticalT_05_2side*stdf_logl
> in)
```

```
// --> Log-Log
gen cilow_price_hat_loglog = exp(ln_price_hat_loglog - criticalT_05_2side*stdf_loglo
> g)

gen cihigh_price_hat_loglog = exp(ln_price_hat_loglog + criticalT_05_2side*stdf_logl
> og)


// (3) Use list to display the confidence intervals at sqft == 2700
disp "95% Confidence Interval for price_hat(sqft = 2700) - Linear: "
95% Confidence Interval for price_hat(sqft = 2700) - Linear:

list cilow_price_hat_lin cihigh_price_hat_lin in `=first_sqft2700'
```

	cil~_lin	cih~_lin
556.	141801	260928.2

```
disp "95% Confidence Interval for price_hat(sqft = 2700) - Log-Linear: "
95% Confidence Interval for price_hat(sqft = 2700) - Log-Linear:

list cilow_price_hat_loglin cihigh_price_hat_loglin in `=first_sqft2700'
```

	cil~glin	cih~glin
556.	133683.1	297315.2

```
disp "95% Confidence Interval for price_hat(sqft = 2700) - Log-Log: "
95% Confidence Interval for price_hat(sqft = 2700) - Log-Log:

list cilow_price_hat_loglog cihigh_price_hat_loglog in `=first_sqft2700'
```

	cilow_~g	cihigh~g
556.	122267	277454.2

```

********************************************************************************
*4.13 Part H: Based on your work in this problem, discuss the choice of functional
* form. Which functional form would you use? Explain.
********************************************************************************

/* Discussion:
> All the models give estimates in roughly the same ball park (large overlaps in
> confidence interval estimates), and all have wide dispersion in the point
> estimates (i.e. relatively wide confidence intervals). The strongest arguments
> for which model to use probably comes from the our analysis of the residuals,
> where the linear model performed very poorly in for high-square footage homes.
> The two log(price) models corrected for this, but between these two the log-
> linear model seemed to do better. Comparing the log-linear model to the log-log
> model, the log-linear has smaller residuals in the high-square footage homes,
> an overall lower sigma_hat^2 estimate, larger t-stat, and higher R2 in both
> the log(price) and price estimates. It is useful to note that it is hard to
> directly compare the regression sigma_hat^2 between the log(price) and price
> regression due to the change in scale caused by using log values.
> */
```

```

*******************************************************************************
*4.13 - Supplementary Figures
*WARNING: This section uses some "advanced STATA" techniques, and has no
*         content related to the class directly.
*******************************************************************************

// Graph to compare fitted values of price. Use only the adjusted values for the
// log-log and log-linear regression

// Have STATA take a picture of the workspace variables that we can return to
// later, undoing any changes that take place between "preserve" and "restore"
preserve


// Create a varlist including all variables with names starting with "price",
// or "cilow" or "cihigh", with * meaning any values (including nothing) after
// the given prefix are allowable.
local price_scale_vars "price* cilow* cihigh*"


// Just to check what variables got put into this local, let's have STATA
// report the names of the variables in this loop. Notice the order of the
// variables (1) the order of the stubs provided, and (2) within each stub, the
// order follows the order in which variables are stored in the workspace
foreach x of varlist `price_scale_vars' {
  2.         disp "`x'"
  3. }
price
price_hat_loglin
price_hat_loglin_adju
price_hat_loglog
price_hat_loglog_adju
price_hat_lin
cilow_price_hat_lin
cilow_price_hat_loglin
cilow_price_hat_loglog
cihigh_price_hat_lin
cihigh_price_hat_loglin
cihigh_price_hat_loglog


// Before plotting, let's adjust the scale of all these variables so that they
// are in terms of thousands of dollars instead of single dollars
foreach x of varlist `price_scale_vars' {
  2.         replace `x' = `x' / 1000
  3. }
(880 real changes made)
(880 real changes made)
(880 real changes made)
(880 real changes made)
(880 real changes made)
(880 real changes made)
(880 real changes made)
(880 real changes made)
(880 real changes made)
(880 real changes made)
(880 real changes made)
(880 real changes made)
```

```

// In addition, let's update the variable labels, since this will be the
// automatic legend labels when we make a figure
label var price "Price - Raw Data"

label var price_hat_lin "Fitted Values: Linear"

label var price_hat_loglin_adju "Fitted Values: Log-Linear"

label var price_hat_loglog_adju "Fitted Values: Log-Log"


// Now, sort by sqft, plot the raw data as a scatter plot, and the fitted
// values for each model as a line
sort sqft

twoway (scatter price sqft) ///
 >                (line price_hat_lin sqft,                       lcolor(red)) ///
 >                (line price_hat_loglin_adju sqft,       lcolor(green)) ///
 >                (line price_hat_loglog_adju sqft,       lcolor(orange)), ///
 >                        ytitle("Price ($ thousands)") xtitle("Square Footage") ///
 >                        title("Price Fitted Values vs. Sqft")

graph export "./Figures/Q 4-13 Price Sqft Fitted Value Lines.pdf", replace
 (file ./Figures/Q 4-13 Price Sqft Fitted Value Lines.pdf written in PDF format)


twoway (scatter price sqft,                                     mcolor(navy)) ///
 >                (line price_hat_lin sqft,                       lcolor(red)) ///
 >                (line price_hat_loglin_adju sqft,       lcolor(green)) ///
 >                (line price_hat_loglog_adju sqft,       lcolor(orange)) ///
 >                (line cilow_*_lin sqft,                         lcolor(red) lpattern
 > ("_")) ///
 >                (line cihigh_*_lin sqft,                        lcolor(red) lpattern
 > ("_")) ///
 >                (line cilow_*_loglin sqft,                      lcolor(green) lpatte
 > rn("_")) ///
 >                (line cihigh_*_loglin sqft,             lcolor(green) lpattern("_"))
 >  ///
 >                (line cilow_*_loglog sqft,                      lcolor(orange) lpatt
 > ern("_")) ///
 >                (line cihigh_*_loglog sqft,             lcolor(orange) lpattern("_")
 > ), ///
 >                        ytitle("Price ($ thousands)") xtitle("Square Footage") ///
 >                        title("Price Fitted Values vs. Sqft") subtitle("(With Confid
 > ence Interals)") ///
 >                        legend(order(1 2 3 4)) text(700 900 "Dashed Lines are Foreca
 > st Confidence Intervals", place(e))

graph export "./Figures/Q 4-13 Price Sqft Fitted Value Lines with CI.pdf", replace
 (file ./Figures/Q 4-13 Price Sqft Fitted Value Lines with CI.pdf written in PDF format
 > )


// Return value of all STATA variables to what they were when I used the
// "preserve" command earlier
restore


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