

Multiple Linear Regression Categorical Predictors & Log Transformations

January 19, 2021
Biostatistics 208

Overview

- Linear regression model for multiple predictors
- Representing multi-level categorical predictors in the linear model
- Post-estimation, contrasts and testing with categorical predictors
 - nominal
 - ordinal
- Interpretation of models with log-transformed outcomes & predictors

Definition of multi-predictor linear regression model

- Linear model for a continuous outcome Y and p predictors:

- mean of Y is linearly related to the x 's:

$$E[y|x_1, x_2, \dots, x_p] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

- individual values of Y are linearly related to the x 's, with normally distributed errors:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \varepsilon$$

- β_0 is the mean of y when all x 's are 0
- β_j is the change in the mean of y associated with a unit increase in x_j , holding the values of all the other x 's fixed (except in complex models, e.g. with interactions)
- Coefficients estimated via least squares
- Coefficients for each predictor account for (or “control for”) the presence of the other predictors

Multilevel Categorical Predictors

- Codes for each level of a multi-level classification
- Nominal example
 - race/ethnicity

```
. codebook raceth
```

```
-----  
raceth                                race/ethnicity  
-----  
  
      type:  numeric (byte)  
      label:  raceth  
  
      range:  [1,3]  
unique values: 3  
  
                                units:  1  
missing .:  0/2763  
  
tabulation:  Freq.    Numeric  Label  
              2451         1  White  
              218         2  African American  
              94          3  Other
```

Multilevel Categorical Predictors (continued)

- Motivation: investigate association between observed glucose levels and race/ethnicity categories in HERS
- Entering a categorical predictor directly in a linear regression model does not give desired results:

```
. regress glucose raceth if diabetes==0
```

Source	SS	df	MS	Number of obs	=	2032
Model	398.460556	1	398.460556	F(1, 2030)	=	4.20
Residual	192619.238	2030	94.8863243	Prob > F	=	0.0406
Total	193017.699	2031	95.0357946	R-squared	=	0.0021
				Adj R-squared	=	0.0016
				Root MSE	=	9.741

glucose	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
raceth	1.140344	.5564744	2.05	0.041	.049024 2.231665
_cons	95.39214	.6555531	145.51	0.000	94.10651 96.67776

- Coefficient estimate for `raceth` gives the change in average glucose associated with a one-unit increase in `raceth`. *This makes no sense!*

Multilevel Categorical Predictors (continued)

- Want the regression model to estimate differences in average glucose between levels of race/ethnicity without assuming linearity of means across categories
- Solution: Create separate binary indicators (dummy variables) for each level of race/ethnicity
`. tabulate raceth, generate(re_cat)`

race/ethnicity	Freq.	Percent	Cum.
White	2,451	88.71	88.71
African American	218	7.89	96.60
Other	94	3.40	100.00
Total	2,763	100.00	

`. regress glucose re_cat* if diabetes==0` *(recall that we wish to exclude diabetic women in this analysis)*

note: re_cat2 omitted because of collinearity

Source	SS	df	MS	Number of obs =	2032
Model	507.191904	2	253.595952	F(2, 2029) =	2.67
Residual	192510.507	2029	94.8795007	Prob > F =	0.0693
Total	193017.699	2031	95.0357946	R-squared =	0.0026
				Adj R-squared =	0.0016
				Root MSE =	9.7406

glucose	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
re_cat1	-.3783718	.9034809	-0.42	0.675	-2.150219 1.393475
re_cat2	0	(omitted)			
re_cat3	2.756824	1.609274	1.71	0.087	-.399178 5.912826
_cons	96.93548	.8747327	110.82	0.000	95.22002 98.65095

Note: Stata only needs 2 indicator variables to estimate the differences in average outcome between the 3 race/eth. categories if the constant (intercept) is included, so it drops one.

Multilevel Categorical Predictors (continued)

- Preceding categorical predictors with `i.` in regression formulae in Stata (versions 11+) automatically creates indicator variables necessary to represent multiple levels of a categorical predictor:

```
. regress glucose i.raceth if diabetes==0
```

Source	SS	df	MS	Number of obs = 2032			
Model	507.191904	2	253.595952	F(2, 2029) = 2.67			
Residual	192510.507	2029	94.8795007	Prob > F = 0.0693			
Total	193017.699	2031	95.0357946	R-squared = 0.0026			
				Adj R-squared = 0.0016			
				Root MSE = 9.7406			

glucose	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
raceth						
African American	.3783718	.9034809	0.42	0.675	-1.393475	2.150219
Other	3.135196	1.369572	2.29	0.022	.4492825	5.821109
_cons	96.55711	.2260983	427.06	0.000	96.1137	97.00052

Note: Stata chooses the lowest numerical level of the categorical variable as the reference category, and omits the indicator variable for this level this from the model

Multilevel Categorical Predictors (continued)

- Definition of indicator variables generated for `raceth` by `i.` with `regress` in Stata:

Category	1.raceth	2.raceth	3.raceth
White (1)	1	0	0
African American (2)	0	1	0
Other (3)	0	0	1

- Resulting regression model:

$$E[\text{glucose} | \text{raceth}] = \beta_0 + \beta_2 2.\text{raceth} + \beta_3 3.\text{raceth}$$

- With constant term in the model, don't need the indicator for the 1st category (taken as the baseline level)
- Coefficient for each indicator gives the difference in mean glucose between that level and the baseline level. For example:

Mean glucose for level 1: β_0

Mean glucose for level 2: $\beta_0 + \beta_2$

Mean glucose for level 3: $\beta_0 + \beta_3$

Difference in mean glucose
between levels 2 & 1: β_2

Difference in mean glucose
between levels 3 & 1: β_3

Multilevel Categorical Predictors (continued)

- Use of `lincom` to get mean glucose for all levels of race/ethnicity:

```
. quietly regress glucose i.raceth if diabetes==0
```

```
. lincom _cons white (1)
```

glucose		Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
(1)		96.55711	.2260983	427.06	0.000	96.1137	97.00052

```
. lincom _cons + 2.raceth african american (2)
```

(1)		96.93548	.8747327	110.82	0.000	95.22002	98.65095
-----	--	----------	----------	--------	-------	----------	----------

```
. lincom _cons + 3.raceth other (3)
```

(1)		99.69231	1.35078	73.80	0.000	97.04325	102.3414
-----	--	----------	---------	-------	-------	----------	----------

- Check results:

```
. tab raceth if diabetes==0, sum(glucose)
```

race/ethnicity		Summary of fasting glucose (mg/dl)		
		Mean	Std. Dev.	Freq.
White		96.557112	9.6006499	1856
African A		96.935484	10.933459	124
Other		99.692308	11.569972	52

Multilevel Categorical Predictors (continued)

- Use of `lincom` to compare mean outcome between levels 3 & 2 of race/ethnicity:

```
. lincom 3.raceth-2.raceth
```

glucose		Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
-----+-----							
(1)		2.756824	1.609274	1.71	0.087	-.399178	5.912826

$$E[\text{glucose} | 3.\text{raceth} = 1] - E[\text{glucose} | 2.\text{raceth} = 1] =$$

$$(\beta_0 + \beta_3) - (\beta_0 + \beta_2) = \beta_3 - \beta_2$$

Tech. note: *Linear Combinations* of Regression Coefficients

- A *linear combination* of $p+1$ coefficients $(\beta_0, \beta_1, \dots, \beta_p)$ from a regression model with p predictors is defined as:

$$\sum_{j=0}^p \alpha_j \beta_j$$

- the α_j terms represent specified numerical constants
- Particular linear combinations arise naturally in interpretation of regression models, and Stata's `lincom` command allows these to be estimated after a model is fitted and for inferences to be made about them
 - Examples include prediction of an individual outcome based on specified predictor values, *contrasts* between estimated mean outcomes between levels of a categorical predictor, and interpretation of interaction effects
- Inferences about linear combinations of coefficients require an estimate of variance (or SE) of the combination, which requires estimates of the individual variances of the involved coefficients and their *covariance* as well.
 - `lincom` standard errors and 95% CIs are based on the following formula for the variance of a linear combination:

$$\text{var}\left(\sum_{j=0}^p \alpha_j \beta_j\right) = \sum_{j=0}^p \alpha_j^2 \text{var}(\beta_j) + 2 \sum_{j=0}^p \sum_{k>0}^p \alpha_j \alpha_k \text{cov}(\beta_j, \beta_k)$$

Linear Combinations of Regression Coefficients - examples

For model considered in the previous example, there are 3 binary predictors representing 3 race/ethnicity categories (the 1st is excluded as the baseline level):

$$E[\text{glucose} | \text{raceth}] = \beta_0 + \beta_2 2.\text{raceth} + \beta_3 3.\text{raceth}$$

The linear combinations for the 4 `lincom` statements above (slides 9 & 10) are :

	α_0	α_2	α_3	$\alpha_0 \beta_0 + \alpha_2 \beta_2 + \alpha_3 \beta_3$
<code>lincom _cons</code>	1	0	0	β_0
<code>lincom _cons + 2.raceth</code>	1	1	0	$\beta_0 + \beta_2$
<code>lincom _cons + 3.raceth</code>	1	0	1	$\beta_0 + \beta_3$
<code>lincom 3.raceth-2.raceth</code>	0	-1	1	$\beta_3 - \beta_2$

Multilevel Categorical Predictors

- Ordinal example
 - grading of physical activity in HERS study (coded 1-5)

```
. codebook physact
```

```
-----  
physact    comparative physical activity  
-----
```

```
      type:  numeric (byte)
      label:  physact
      range:  [1,5]
unique values: 5
      units:  1
      missing .: 0/2763
      tabulation: Freq.    Numeric    Label
                  197      1    much less active
                  503      2    somewhat less active
                  919      3    about as active
                  838      4    somewhat more active
                  306      5    much more active
```

Multilevel Categorical Predictors (continued)

- Regression model for dependence of glucose on an ordinal categorical measure:

```
. regress glucose physact if diabetes==0
```

Source	SS	df	MS	Number of obs = 2032		
Model	1598.22653	1	1598.22653	F(1, 2030) = 16.95		
Residual	191419.472	2030	94.2953065	Prob > F = 0.0000		
Total	193017.699	2031	95.0357946	R-squared = 0.0083		
				Adj R-squared = 0.0078		
				Root MSE = 9.7106		
glucose	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
physact	-.8465276	.2056208	-4.12	0.000	-1.249777	-.4432778
_cons	99.46872	.7153373	139.05	0.000	98.06585	100.8716

- Model excludes individuals with diabetes
- Coefficient for physact gives the change in average glucose associated with a one-unit increase in physact
- physact is not a continuous measure, and we usually do not want to assume that glucose changes in the same way between all levels of activity

Multilevel Categorical Predictors (continued)

- Using `i.` syntax in regress:

```
. regress glucose i.physact if diabetes==0
```

Source	SS	df	MS
Model	1673.09022	4	418.272554
Residual	191344.609	2027	94.3979322
Total	193017.699	2031	95.0357946

```
Number of obs =      2032
F(   4,  2027) =      4.43
Prob > F       =     0.0014
R-squared      =     0.0087
Adj R-squared  =     0.0067
Root MSE      =     9.7159
```

glucose	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
physact						
somewhat less active	-.8584489	1.084152	-0.79	0.429	-2.984617	1.267719
about as active	-1.226199	1.011079	-1.21	0.225	-3.20906	.7566629
somewhat more active	-2.433855	1.010772	-2.41	0.016	-4.416114	-.451595
much more active	-3.277704	1.121079	-2.92	0.003	-5.476291	-1.079116
_cons	98.42056	.9392676	104.78	0.000	96.57853	100.2626

Multilevel Categorical Predictors (continued)

- Using `b( )` syntax to specify baseline category (in this case “5: much more active”):

```
. regress glucose ib(5).physact if diabetes==0
```

Source	SS	df	MS	Number of obs = 2032			
Model	1673.09022	4	418.272554	F(4, 2027) = 4.43			
Residual	191344.609	2027	94.3979322	Prob > F = 0.0014			
Total	193017.699	2031	95.0357946	R-squared = 0.0087			
				Adj R-squared = 0.0067			
				Root MSE = 9.7159			

glucose		Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
physact							
much less active		3.277704	1.121079	2.92	0.003	1.079116	5.476291
somewhat less active		2.419255	.8171635	2.96	0.003	.8166866	4.021823
about as active		2.051505	.717392	2.86	0.004	.6446024	3.458407
somewhat more active		.8438489	.7169593	1.18	0.239	-.562205	2.249903
_cons		95.14286	.6120416	155.45	0.000	93.94256	96.34315

Multilevel Categorical Predictors (continued)

- Definition of indicator variables generated for `physact` by `i.` with `regress`:

Physical activity category	1.physact	2.physact	3.physact	4.physact	5.physact
much less active (1)	1	0	0	0	0
somewhat less active (2)	0	1	0	0	0
about as active (3)	0	0	1	0	0
somewhat more active (4)	0	0	0	1	0
much more active (5)	0	0	0	0	1

- Resulting regression model:

$$E[\text{glucose} | \text{physact}] = \beta_0 + \beta_2 2.\text{physact} + \beta_3 3.\text{physact} + \beta_4 4.\text{physact} + \beta_5 5.\text{physact}$$

- With constant term in the model, don't need the indicator for the 1st category (taken as the baseline level)
- Coefficient for each indicator gives the difference in mean glucose between that level and the baseline level. For example:

Mean glucose for level 2: $\beta_0 + \beta_2$

Mean glucose for level 1: β_0

Difference in mean glucose
between levels 2 & 1: β_2

Multilevel Categorical Predictors (continued)

- Use of `lincom` to get mean glucose for all levels of physical activity:

```
. quietly regress glucose i.physact if diabetes==0
```

```
. lincom _cons much less active (1)
```

glucose	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
(1)	98.42056	.9392676	104.78	0.000	96.57853	100.2626

```
. lincom _cons + 2.physact somewhat less active (2)
```

(1)	97.56211	.5414437	180.19	0.000	96.50027	98.62396
-----	----------	----------	--------	-------	----------	----------

```
. lincom _cons + 3.physact about as active (3)
```

(1)	97.19436	.3742409	259.71	0.000	96.46043	97.9283
-----	----------	----------	--------	-------	----------	---------

```
. lincom _cons + 4.physact somewhat more active (4)
```

(1)	95.98671	.3734108	257.05	0.000	95.2544	96.71902
-----	----------	----------	--------	-------	---------	----------

```
. lincom _cons + 5.physact much more active (5)
```

(1)	95.14286	.6120416	155.45	0.000	93.94256	96.34315
-----	----------	----------	--------	-------	----------	----------

Multilevel Categorical Predictors (continued)

- The linear model including binary indicators to represent levels of a categorical predictor allows comparisons between any desired levels of physical activity
- Use of `lincom` to get difference in mean glucose between two specified levels of physical activity (e.g. level 5 compared to level 3):

Mean glucose for level 5: $\beta_0 + \beta_5$

Mean glucose for level 3: $\beta_0 + \beta_3$

Difference: $\beta_5 - \beta_3$

```
. lincom 5.physact - 3.physact
```

```
( 1)  - 3.physact + 5.physact = 0
```

glucose	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
-----+-----						
(1)	-2.051505	.717392	-2.86	0.004	-3.458407	-.6446024

Multilevel Categorical Predictors (continued)

- Testing overall impact of physical activity as a predictor of average glucose level in HERS
- F statistic in the standard output tests the *homogeneity* hypothesis that the true values for all of the (4) coefficients used to represent physical activity are equal to zero

```
. regress glucose i.physact if diabetes==0
```

Source	SS	df	MS	Number of obs =	2032
-----+-----				F(4, 2027) =	4.43
Model	1673.09022	4	418.272554	Prob > F =	0.0014
Residual	191344.609	2027	94.3979322	R-squared =	0.0087
-----+-----				Adj R-squared =	0.0067
Total	193017.699	2031	95.0357946	Root MSE =	9.7159

Rest of output omitted ...

- The `testparm` command can be used to evaluate this, even when additional predictors are included in the model:

```
. testparm i.physact
```

```
( 1) 2.physact = 0
( 2) 3.physact = 0
( 3) 4.physact = 0
( 4) 5.physact = 0
```

```
F( 4, 2027) = 4.43
Prob > F = 0.0014
```

Multilevel Categorical Predictors (continued)

- F test allows an overall test of the contribution of `physact` to the model by evaluating evidence for heterogeneity in group-specific means
- When only a single categorical predictor is included, results are identical to one-way ANOVA:

. anova glucose physact if diabetes==0

	Number of obs = 2032			R-squared = 0.0087	
	Root MSE = 9.71586			Adj R-squared = 0.0067	
Source	Partial SS	df	MS	F	Prob > F
Model	1673.09022	4	418.272554	4.43	0.0014
physact	1673.09022	4	418.272554	4.43	0.0014
Residual	191344.609	2027	94.3979322		
Total	193017.699	2031	95.0357946		

- individual t -tests only allow evaluation of each activity level to the baseline level
 - Note: multiple tests harder to interpret, and can multiple testing inflate experiment-wise error rate (EER , see section 4.3.4 of textbook)

Multilevel Categorical Predictors (continued)

- Testing overall impact of physical activity as a predictor of average glucose level in HERS
- The `test` command can be also used to evaluate the hypothesis of no effect of physical activity:

```
. test 2.physact 3.physact 4.physact 5.physact
```

```
( 1) 2.physact = 0
```

```
( 2) 3.physact = 0
```

```
( 3) 4.physact = 0
```

```
( 4) 5.physact = 0
```

```
F( 4, 2027) = 4.43  
Prob > F = 0.0014
```

- **Note:** `test` & `testparm` are related commands:
 - `test` expects coefficient names as arguments
 - `testparm` accepts variable names as arguments, accepts some different options
 - For the linear regression model, results are evaluated using *F*-tests

Multilevel Categorical Predictors (continued)

- The test performed by `testparm` & `test` in the previous slides is an example of using the F test to compare *nested* models.

1. Fit the *full model* including all the variable(s) of interest. Call this model 1.

```
. quietly regress glucose i.physact if diabetes==0
```

2. Fit the *nested model* excluding the variable(s) of interest. Call this model 2:

```
. regress glucose if diabetes==0
```

Source	SS	df	MS	Number of obs	=	2032
-----+-----				F(0, 2762)	=	0.00
Model	0	0	.	Prob > F	=	.
Residual	193017.699	2031	95.0357946	R-squared	=	0.0000
-----+-----				Adj R-squared	=	0.0000
Total	193017.699	2031	95.0357946	Root MSE	=	9.7486

Rest of output omitted ...

3. Compute the following statistic comparing the residual sums of squares (RSS) and corresponding degrees of freedom between the two models:

$$F = [(RSS_2 - RSS_1) / (df_2 - df_1)] / (RSS_1 / df_1)$$

(this statistic has the F distribution with $(df_2 - df_1, df_2)$ degrees of freedom

```
. dis ((193017.699 - 191344.609)/(2031-2027)) / (191344.609/2027)
4.4309498 (F-statistic)
. dis 1-F(4,2031,4.4309498)
.00144076 (p-value)
```

4. *Note*: Don't do this by hand in practice. It can be more easily accomplished using the `testparm` or `test` commands as shown above

Multilevel Categorical Predictors - Contrasts

- A *contrast* of regression coefficients is a weighted sum of coefficients where the weights (aka *contrast coefficients*) sum to zero
 - Contrasts are special cases of linear combinations
- Contrasts can be used to evaluate hypotheses comparing levels of a categorical predictor

Examples:

- Differences between levels of a categorical predictor
- Comparisons of level-specific effects with a reference level
- *Trend* across levels of an ordinal categorical predictor

Multilevel Categorical Predictors (continued)

- Use of `contrast` to get difference in mean glucose between two specified levels of physical activity (e.g. level 5 compared to level 3 as in slide 19):

```
. contrast {physact 0 0 -1 0 1}, effects
```

Contrasts of marginal linear predictions

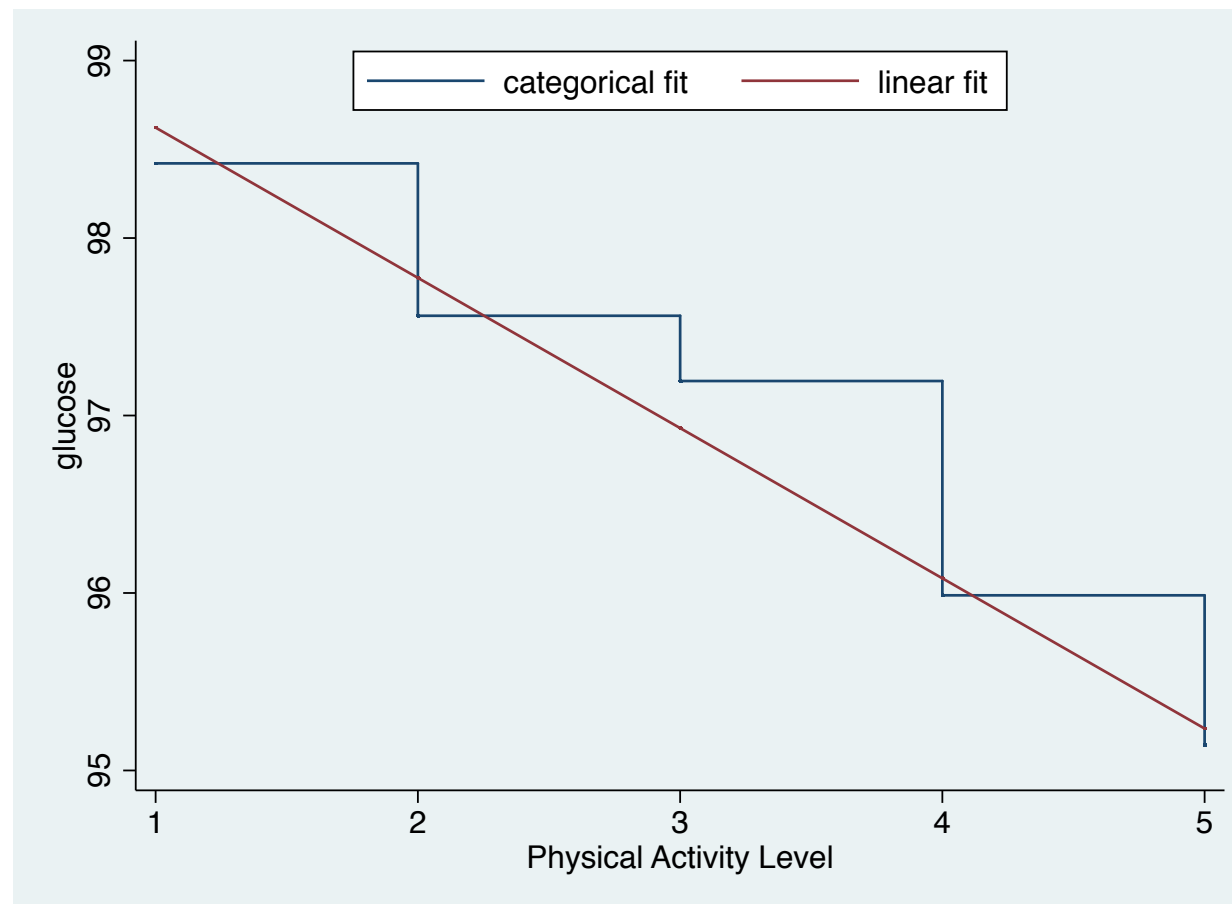
Margins : asbalanced

	df	F	P>F
physact	1	8.18	0.0043
Residual	2027		

	Contrast	Std. Err.	t	P> t	[95% Conf. Interval]	
physact (1)	-2.051505	.717392	-2.86	0.004	-3.458407	-.6446024

Testing for linear trend

- Above results indicate that average glucose varies across levels of `physact`
- Since `physact` can be viewed as an ordinal categorical variable, assessing trend in average glucose with increasing activity levels is appropriate
- Trend can be evaluated via the slope of a line fitted to category-specific mean outcome levels
- A significant test for trend indicates that the line has non-zero slope



- Problem with trend testing based on fitting a line that treating the category labels as a continuous predictor: labeling of categories is often arbitrary

Trend Test: example

- Linear trend can be evaluated explicitly using the `contrast` command

```
. quietly regress glucose i.physact if diabetes==0  
. contrast {physact -2 -1 0 1 2}, noeffects
```

Contrasts of marginal linear predictions

Margins	:	asbalanced		
		df	F	P>F
	+			
physact		1	12.11	0.0005
Denominator		2027		

- Evaluates the null hypothesis that the slope describing a linear trend in the category-specific means is zero
- Motivated by the slope coefficient for a linear regression fitted to the category-specific means, using the integer category labels as the predictor
- Contrast coefficients satisfy the following properties
 - integer-valued
 - evenly spaced
 - symmetric about zero

Trend Test: example

- Trend can also be evaluated using the `test` command (equivalent to `contrast` as used above)
- The following `test` command applied after the model including `physact` is fitted evaluates the significance of linear trend using the contrast coefficients explicitly:

```
. quietly regress glucose i.physact if diabetes==0  
. test -2.physact + 4.physact + 2*5.physact=0
```

```
( 1)  - 2.physact + 4.physact + 2*5.physact = 0
```

```
      F( 1, 2027) =    12.11  
      Prob > F =    0.0005
```

- The contrast used in `test` omits the coefficient for the first category (`1.physact`) because this isn't used in the fitted model (i.e. the model includes an intercept term)
- Conclude that there is evidence for a linear trend in mean glucose across levels of `physact`

Trend Test: motivation for contrast coefficients

- Where do the coefficients $(-2 \ -1 \ 0 \ 1 \ 2)$ come from in the above `contrast` command?
 - `physact` has 5 levels, so coefficients that are integer-valued, evenly spaced and symmetric about zero are $(-2 \ -1 \ 0 \ 1 \ 2)$
 - e.g. for a 6-level categorical predictor: $(-5 \ -3 \ -1 \ 1 \ 3 \ 5)$
- In models including an intercept term, the coefficient from the base category isn't used when trend is evaluated with the `test` command (previous slide):

Number of levels	Model Coefficients	Linear contrast in coefficients
3	β_2, β_3	$\beta_3 = 0$
4	$\beta_2, \beta_3, \beta_4$	$-\beta_2 + \beta_3 + 3\beta_4 = 0$
5	$\beta_2, \beta_3, \beta_4, \beta_5$	$-\beta_2 + \beta_4 + 2\beta_5 = 0$
6	$\beta_2, \beta_3, \beta_4, \beta_5, \beta_6$	$-3\beta_2 - \beta_3 + \beta_4 + 3\beta_5 + 5\beta_6 = 0$

See Table 4.8 in section 4.3.5 of textbook

For a more technical explanation, there is a handout posted on the course web site.

Trend Test: example

- Evidence for the presence of linear trend can be *most easily be assessed* using the `contrast` command and the `q` operator:

```
. contrast q(1).physact, noeffects
```

Contrasts of marginal linear predictions

Margins : asbalanced

	df	F	P>F
physact	1	12.11	0.0005
Residual	2027		

- **Recommendation:** use the `contrast` command as illustrated on this slide for evaluating linear trend of an ordinal categorical predictor
- `q(1)` option requests linear trend, and automatically uses the appropriate contrast
- works only for categories with *equal spacings* (see section 4.3.5 for alternatives)
- The null hypothesis is that the slope of the linear trend is zero - a significant *p*-value indicates evidence for linear trend

Trend Test: evaluating evidence for departure from linear trend

- Trend in levels of an ordinal categorical predictor can be characterized by presence of a linear component as well as a nonlinear part
- Testing for *departure from linear trend* allows evaluation of this (after presence of a trend is established)
- Can also use contrast command (see section 4.3.5.1 for details):

```
. contrast q(2/4).physact, noeffects
```

Contrasts of marginal linear predictions

Margins : asbalanced

	df	F	P>F
physact			
(quadratic)	1	0.11	0.7411
(cubic)	1	0.01	0.9415
(quartic)	1	0.49	0.4859
Joint	3	0.26	0.8511
Denominator	2027		

- requests additional tests based on evaluating quadratic, cubic and quartic trend components (omitting linear component)
- `q(2/4)` option is based on number of levels in `physact` used in model (minus 1)

Results

- We see that there is a pattern of decrease in glucose with physical activity
- p -value for trend test is highly significant; consistent with the presence of a linear trend
- *Note* : evaluating trend by including a numerically coded categorical predictor as a continuous score in a regression model, and using the resulting regression coefficient and associated t -test is generally not equivalent (and not preferred) to the more general approach outlined in the past few slides

Multilevel Categorical Predictors (continued)

Homogeneity Between Outcome Means for Levels of a Categorical Predictor vs. Trend in Mean over Levels:

- Homogeneity: no difference in mean outcome between the groups coded by the categorical predictor
- Trend: difference between groups which implies a trend in mean glucose with increasing levels of the categorical predictor
- “Trend” only makes sense if predictor variable is ordered in some way
- Trend test can be based on a specified combination of coefficients (a so-called *contrast*) and tested using `test` or `contrast` commands in Stata (section 4.3.5 of textbook)
 - The “`contrast q(1)`” approach for a linear trend test (above) is *recommended* as the simplest

Interpreting results for log-transformed variables

- Positive continuous variables are frequently log-transformed in regression models
 - *outcomes*: normalize and equalize variance
 - *predictors*: address non-linearity or interaction issues, reduce influence of outliers
- Both natural log ($\ln$) and $\log_{10}$ ($\log_{10}$) transformations frequently used
- How does this affect interpretation of regression coefficients?
- For a more detailed description of the material covered here, see:

<http://www.emersonstatistics.com/GeneralMaterials/Ratios%20and%20Logarithms.pdf>

Brief review of exponentials and natural logarithms

Properties of exponentials

$$e = 2.7182818\dots$$

$$e^a = \exp(a)$$

$$e^a e^b = e^{a+b}, e^a / e^b = e^{a-b}$$

$$(e^a)^b = e^{ab}$$

$$e^0 = 1$$

$$-\infty < a \leq 0, \text{ then } 0 < e^a \leq 1$$

$$0 < a < \infty, \text{ then } 1 < e^a < \infty$$

Properties of logarithms

$$\log(e) = 1$$

$$\log(e^a) = a, \exp(\log(a)) = a$$

$$\log(ab) = \log(a) + \log(b), \log(a/b) = \log(a) - \log(b)$$

$$\log(a^b) = b \log(a)$$

$$\log(1) = 0$$

$$0 < a \leq 1, \text{ then } -\infty < \log(a) \leq 0$$

$$1 < a < \infty, \text{ then } 0 < \log(a) < \infty$$

Similar properties apply to logs of other bases (e.g. $\log_{10}(10^a) = a$, $10^{\log_{10}(a)} = a$)

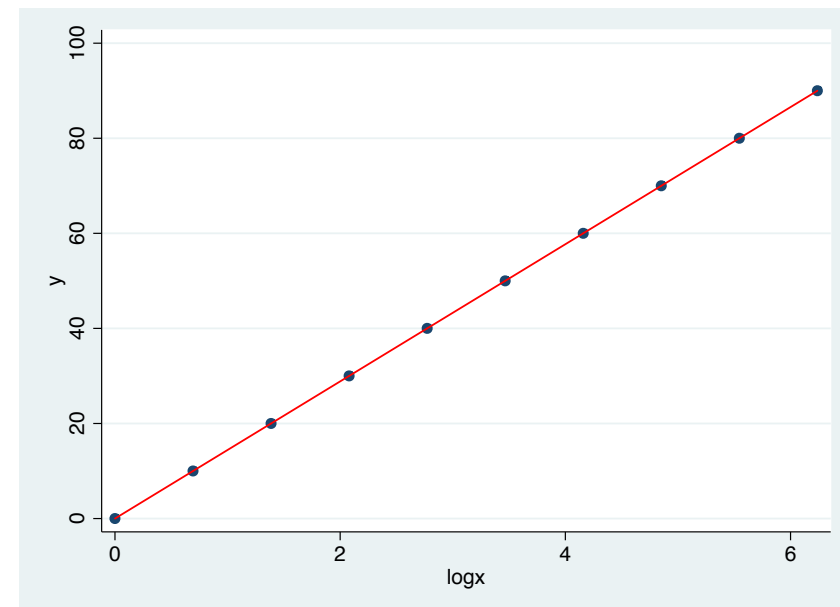
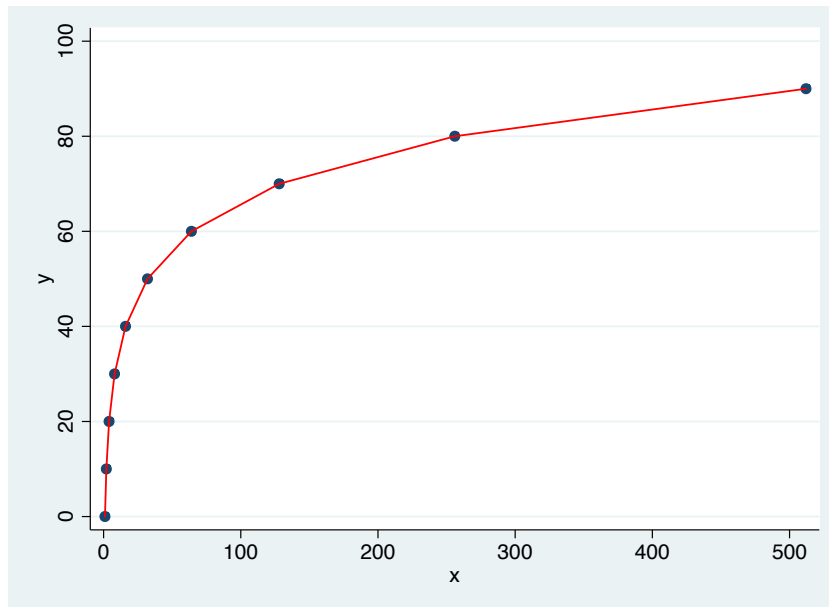
Log-transformed predictors

- We'll assume that the natural logarithm (i.e. log to the base e) of a variable x is denoted $\log(x)$, or equivalently, $\ln(x)$
 - log to another base (e.g. 10) will be denoted $\log_{10}(x)$
- Example: models for diminishing response (y) with increasing dose (x):

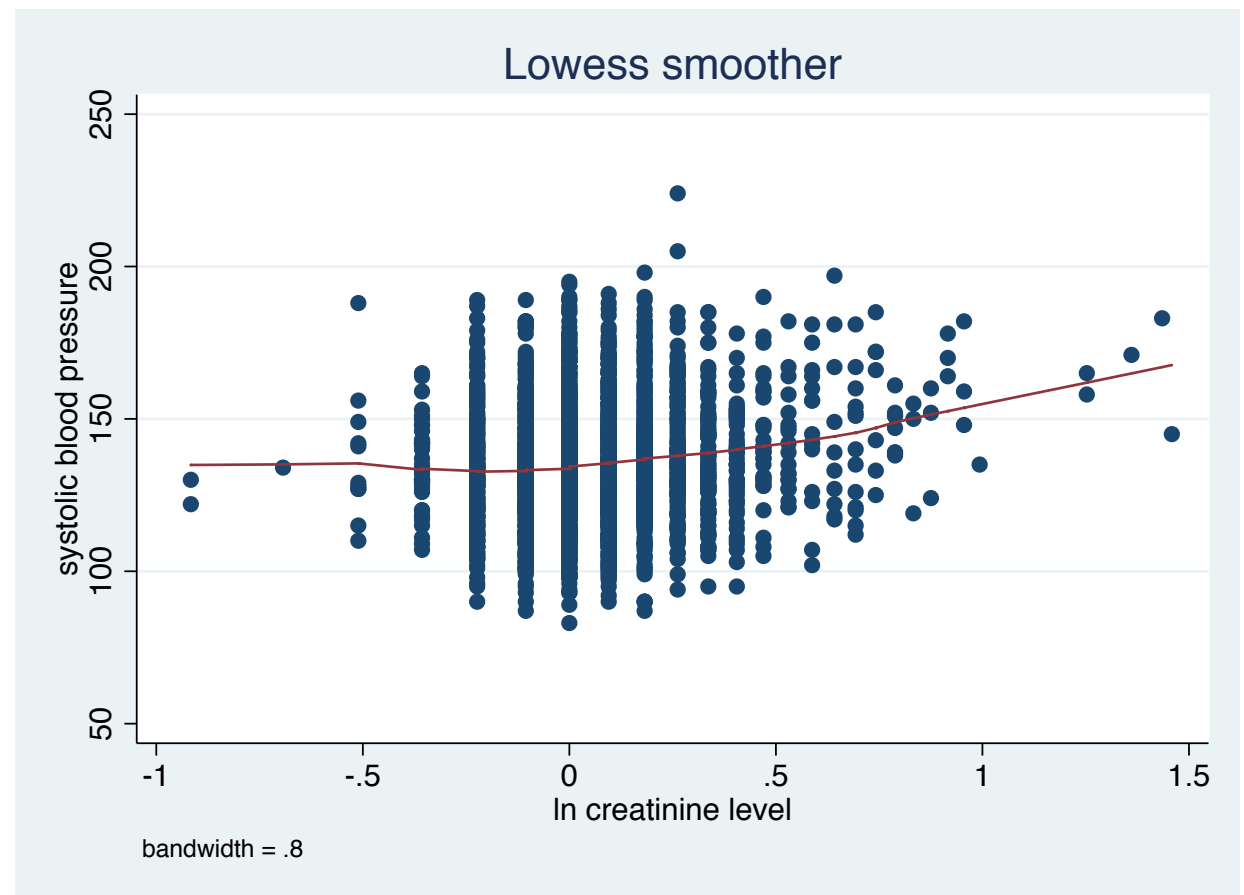
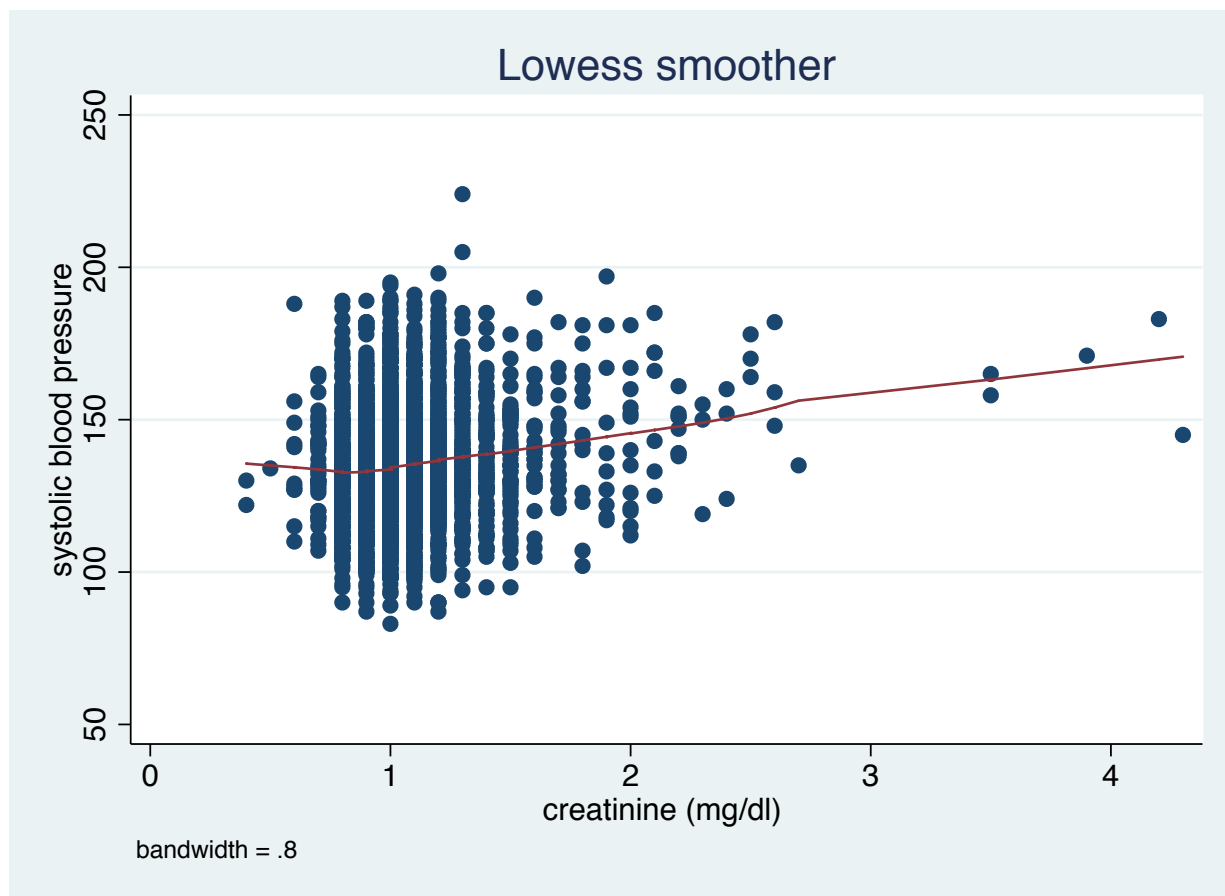
$$E[y|x] = \beta_0 + \beta_1 \log(x)$$

- For example, $\beta_0 = 0$, $\beta_1 = 10/\log(2)$ models a 10-unit increase in $E[y|x]$ for each doubling of x

x	$E[y x]$
1	0
2	10
4	20
8	30
16	40



Example: SBP and log-transformed creatinine in HERS



Log transformation (right) of creatinine serves to “pull in” outlying values

Log-transformed predictors

$$E[y|x] = \beta_0 + \beta_1 \log(x)$$

- In a linear regression model with a natural log-transformed predictor x , β_1 gives the increase in $E[y|x]$ for a one-unit increase in $\log(x)$
 - This corresponds to an e -fold ($e=2.718..$) increase in x
i.e. $\log(x)+1 = \log(x) + \log(e) = \log(x \times e)$
- If predictor x is $\log_{10}$ -transformed, β_1 gives the increase in $E[y|x]$ for a one-unit increase in $\log_{10}(x)$
 - This corresponds to an 10-fold increase in x
i.e. $\log_{10}(x)+1 = \log_{10}(x) + \log_{10}(10) = \log_{10}(x \times 10)$
- How can we get more interpretable estimates?

Log-transformed predictors

- For a log-transformed predictor x

$$E[y|x] = \beta_0 + \beta_1 \log(x)$$

- change in mean outcome assoc. with a k - fold *multiplicative increase* in x :

$$[\beta_0 + \beta_1 \log [x \times k]] - [\beta_0 + \beta_1 \log(x)] = \beta_1 \log \left(\frac{x \times k}{x} \right) = \beta_1 \log (k)$$

doesn't depend on the value of x

- change in mean outcome assoc. with a k - unit *additive increase* in x :

$$[\beta_0 + \beta_1 \log [x + k]] - [\beta_0 + \beta_1 \log(x)] = \beta_1 \log \left(\frac{x + k}{x} \right) = \beta_1 \log \left(1 + \frac{k}{x} \right)$$

depends on the value of x

Log-transformed predictors

- For a natural log-transformed predictor x

$$E[y|x] = \beta_0 + \beta_1 \log(x)$$

- $\beta_1 \log(1 + k/100)$: gives change in $E[y|x]$ for a $k\%$ increase in x :

$$[\beta_0 + \beta_1 \log[x(1 + k/100)]] - [\beta_0 + \beta_1 \log(x)] = \beta_1 \log\left(\frac{x(1 + k/100)}{x}\right) = \beta_1 \log(1 + k/100)$$

- For a $\log_{10}$ -transformed predictor x

$$E[y|x] = \beta_0 + \beta_1 \log_{10}(x)$$

- $\beta_1 \log_{10}(1 + k/100)$: gives change in $E[y|x]$ for a $k\%$ increase in x

- *Note:*

- Same interpretation holds for models with multiple predictors
 - Must use `nlcom` to get estimates with confidence intervals

- **Note:** `nlcom` provides estimates and confidence intervals for *nonlinear combinations* of coefficients (i.e. combinations that can't be estimated using `lincom`)

Example: SBP and log-transformed creatinine in HERS

```
. regress sbp lncreat age diabetes
```

Source	SS	df	MS	Number of obs =	2761
Model	62724.9665	3	20908.3222	F(3, 2757) =	61.70
Residual	934341.036	2757	338.897728	Prob > F =	0.0000
Total	997066.003	2760	361.255798	R-squared =	0.0629
				Adj R-squared =	0.0619
				Root MSE =	18.409

sbp	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
lncreat	9.211717	1.682269	5.48	0.000	5.913081	12.51035
age	.444083	.0536661	8.27	0.000	.3388531	.5493129
diabetes	6.426345	.8007529	8.03	0.000	4.856209	7.996481
_cons	103.2765	3.600996	28.68	0.000	96.21558	110.3374

```
. nlcom _b[lncreat]*log(1 + 0.5)
```

```
_nl_1: _b[lncreat]*log(1 + 0.5)
```

sbp	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_nl_1	3.73503	.6821016	5.48	0.000	2.398135	5.071924

Interpretation: average SBP increases 3.7 mmHg with each 50% increase in creatinine (95% CI: 2.4, 5.1, $p < 0.001$)

Note: nlcom is used for nonlinear combinations of coefficients

- A *nonlinear combination* of $p+1$ coefficients $(\beta_0, \beta_1, \dots, \beta_p)$ from a regression model can't be expressed as a linear combination as defined above (slide 11)
 - Examples include products or ratios involving coefficients, or expressions involving nonlinear functions of coefficients (e.g. $\alpha \times \exp(\beta_1)$; $\alpha + \log(\beta_1)$)
- Stata's nlcom command allows these to be estimated after a model is fitted and for inferences to be made about them
- Inferences about nonlinear combinations of coefficients require an estimate of variance (or SE) of the combination, which requires estimates of the individual variances of the involved coefficients and their *covariance* as well.
 - nlcom standard errors and 95% CIs are based on an approximate variance estimate that is generally less accurate than that provided by lincom
- **Advice:** Use lincom, unless you need nlcom!

Natural log-transformed outcomes

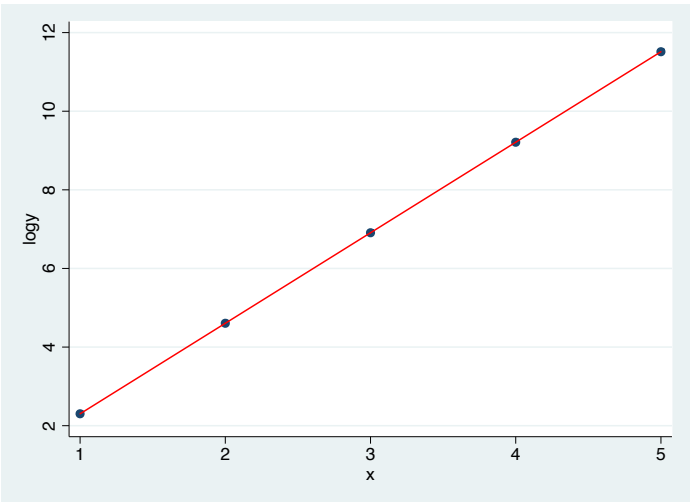
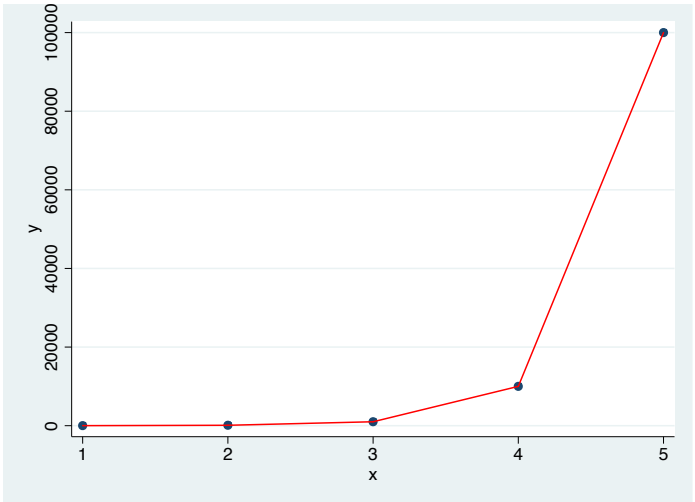
- Models increasing or decreasing exponential dose-response

$$E[\log(y)|x)] \approx \log[E(y|x)] = \beta_0 + \beta_1 x$$

- For example, $\beta_0 = 0$, $\beta_1 = \pm \log(10)$ models a 10-fold (relative) change in $E[y|x]$ for each unit increase in x

E[y x]		
x	$\beta_1 = \log(10)$	$\beta_1 = -\log(10)$
1	10	0.1
2	100	0.01
3	1000	0.001
4	10000	0.0001
5	100000	0.00001

For $\beta_1 = \log(10)$:



- Interpretation is *approximate*: $E[\log(y)] \neq \log(E[y])$

Natural log-transformed outcomes

- In a linear regression model with a natural log-transformed outcome y ,
 - β_1 gives the *additive* increase in $E[\log(y)|x]$ for a one-unit increase in x
 - $\exp(\beta_1) = e^{\beta_1}$ gives the approximate *relative* increase in $E[y|x]$ for a one-unit increase in x (e.g. if $e^{\beta_1} = 1.5$, a 1.5-fold increase in $E[y|x]$ for a 1-unit increase in x)
 - similar result holds with bases other than e (e.g. use 10^{β_1} for base 10 logs)
- The *percent increase* in $E[y|x]$ for a unit increase in x is given by $100(e^{\beta_1} - 1)$
 - i.e. $e^{\beta_1} = 1.5$ represents a 50% increase in $E[y|x]$ for each 1-unit increase in x
- *Note:* adding a small amount to outcomes makes it possible to apply log transformation to a variable that is zero for some observations (e.g. $\log(y+1)$)

Natural log-transformed outcomes

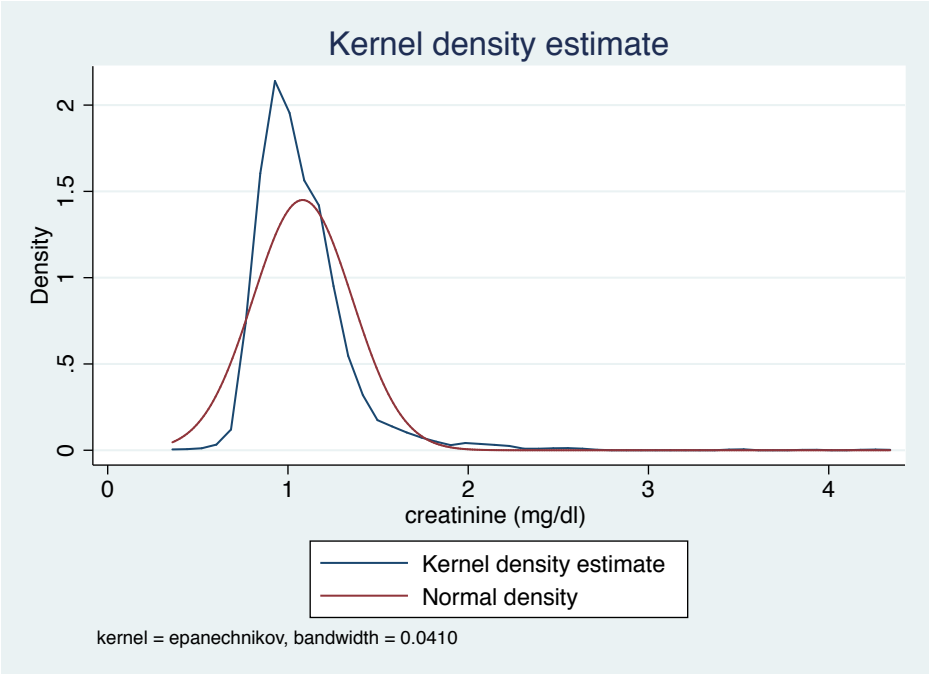
- For untransformed predictors:
 - use option `eform( "exp(beta)" )` to have `regress` display output giving relative effects e^{β_1}
 - use `lincom` with `eform` option to get relative increase in (untransformed) outcome per k -unit increase in predictor (where k is an arbitrary integer)
 - use `nlcom` to get percent increase in outcome per k -unit increase in predictor

Example: effect of age on log-transformed creatinine in HERS

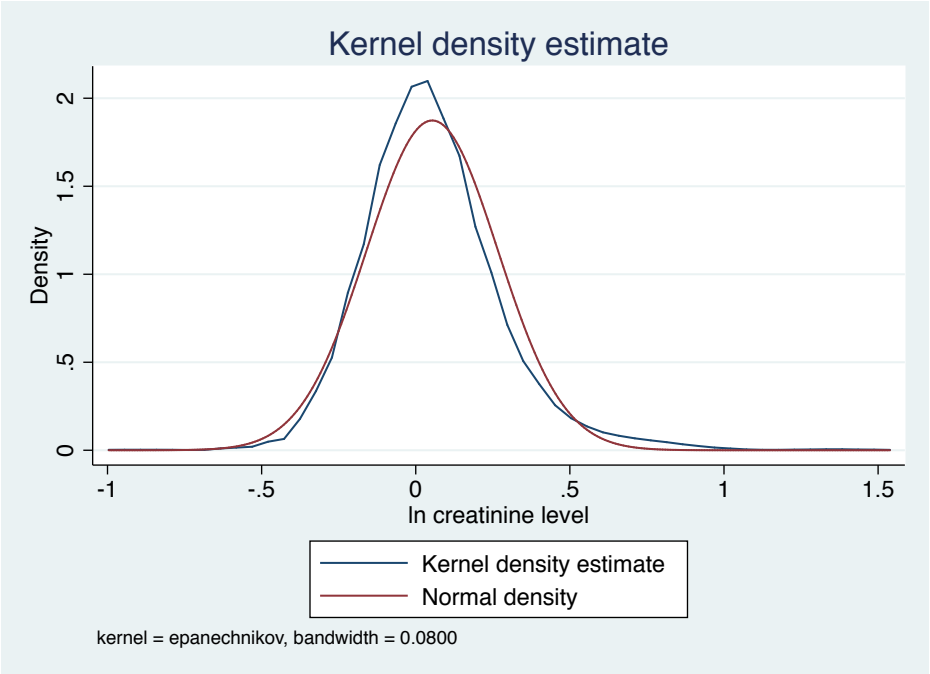
```
. regress lncreat bmi age, eform("exp(beta)")
```

Source	SS	df	MS	Number of obs = 2756	
Model	4.76569402	2	2.38284701	F(2, 2753) =	55.00
Residual	119.265116	2753	.043321873	Prob > F =	0.0000
				R-squared =	0.0384
				Adj R-squared =	0.0377
Total	124.03081	2755	.045020258	Root MSE =	.20814

lncreat	exp(beta)	Std. Err.	t	P> t	[95% Conf. Interval]	
bmi	1.003308	.0007303	4.54	0.000	1.001877	1.004741
age	1.006093	.0006076	10.06	0.000	1.004902	1.007285
_cons	.6405172	.0309547	-9.22	0.000	.5826076	.704183



untransformed



log-transformed

Example: effect of age on log-transformed creatinine in HERS

* relative increment in creatinine associated with 10-year increase in age

. **lincom** age*10, **eform**

(1) 10*age = 0

lncreat	exp(b)	Std. Err.	t	P> t	[95% Conf. Interval]	
(1)	1.062624	.0064171	10.06	0.000	1.050115	1.075282

- Creatinine increases 1.063-fold for each 10-year increase in age

. **nlcom** 100*(exp(_b[age]*10)-1)

_nl_1: 100*(exp(_b[age]*10)-1)

lncreat	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_nl_1	6.262402	.6417065	9.76	0.000	5.00468	7.520123

- Creatinine increases 6.3% for each 10-year increase in age

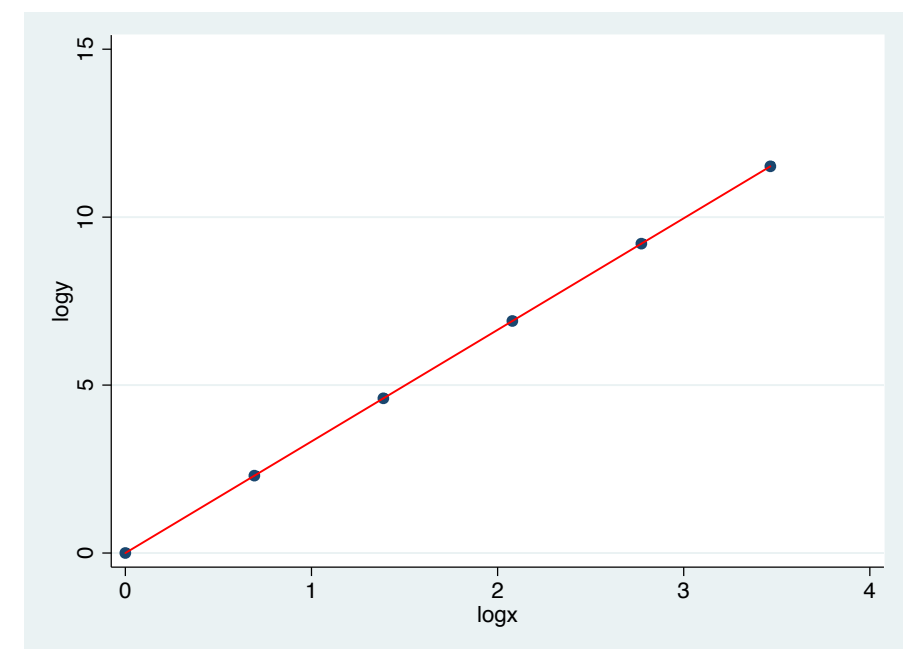
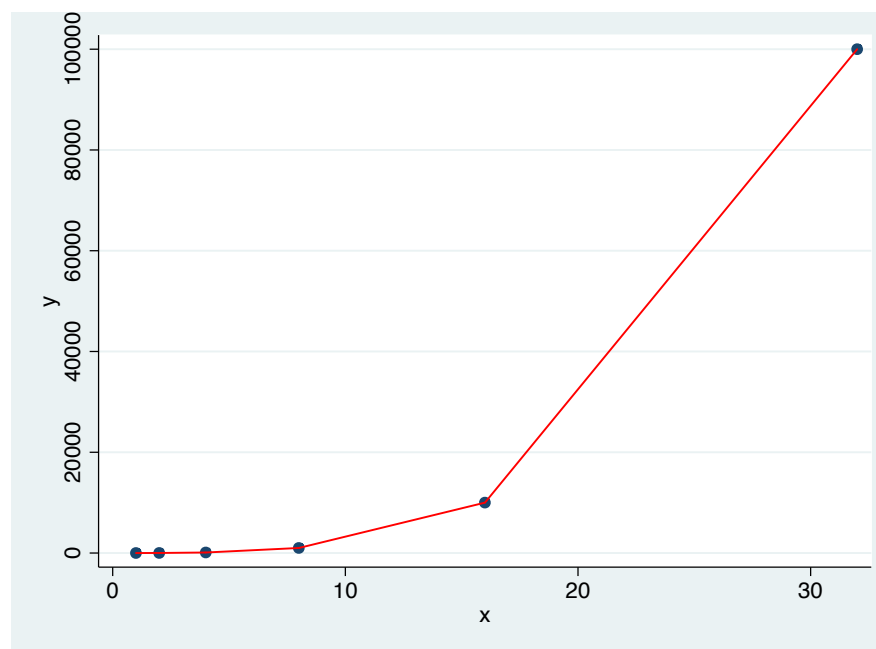
Outcome & predictor both log-transformed

- Models exponential response to proportional increases in dose

$$E[\log(y)|x] = \beta_0 + \beta_1 \log(x)$$

- For example, $\beta_1 = \log(10)/\log(2)$ (with $\beta_0 = 0$) models a 10-fold (relative) change in $E[y|x]$ for each doubling of x

x	$E[y x]$
1	1
2	10
4	100
8	1000
16	10000



Outcome & predictor both log-transformed

- In a linear regression model with a natural log-transformed predictor x and outcome y :
 - $\exp(\beta_1 \log(1 + k/100))$ gives the *relative* increase in $E[y|x]$ for a $k\%$ increase in x
 - $100 (\exp(\beta_1 \log(1 + k/100)) - 1)$ gives the *percent* increase in $E[y|x]$ for a $k\%$ increase in x
- use `eform` with `regress`, `nlcom` for estimates with CIs
- See VGSM, Sect. 4.7.5 for more details - examples in lab this week

Example: Log-transformed creatinine and TG in HERS

```
. regress lncreat bmi age lntgl, eform("exp(beta)")
```

Source	SS	df	MS	Number of obs = 2752			
-----+-----				F(3, 2748) = 42.15			
Model	5.45380764	3	1.81793588	Prob > F = 0.0000			
Residual	118.530536	2748	.043133383	R-squared = 0.0440			
-----+-----				Adj R-squared = 0.0429			
Total	123.984344	2751	.045068827	Root MSE = .20769			

lncreat	exp(beta)	Std. Err.	t	P> t	[95% Conf. Interval]		
-----+-----							
bmi	1.002752	.0007412	3.72	0.000	1.0013	1.004206	
age	1.006025	.0006069	9.96	0.000	1.004835	1.007216	
lntgl	1.041705	.010456	4.07	0.000	1.021403	1.06241	
_cons	.5321261	.0353937	-9.48	0.000	.4670604	.606256	

```
. nlcom 100*(exp(_b[lntgl]*log(1.25))-1)
```

_nl_1: 100*(exp(_b[lntgl]*log(1.25))-1)						

lncreat		Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
-----+						
_nl_1		.9159048	.2260289	4.05	0.000	.4728963 1.358913

Creatinine increases 0.9% for each 25% increase in TG

Summary

- Multiple linear regression model extends simple linear model and involves similar assumptions
- Categorical predictors factored into binary indicators (use `i.` with `regress` in Stata)
- Default coding: coefficients represent mean outcome difference for successive levels compared to a reference level
- Can perform: overall F test, trend test or examine pairwise comparisons
- Trend tests useful with ordinal categorical variables
- Interpretation of coefficients from models involving log-transformed outcomes/predictors