POL 681 Lecture Notes
Fun With Dummy Variables
(Use in conjunction with Lecture Notes)

PART I: Preliminaries

Let’s use the dataset nesexample.dta and estimate a model with a dummy variable. The dummy variable codes whether or not a respondent was Latino (1) or non-Latino (0). It is referred to as “hispan” in the data set below. Here is a summary table:

```
. table hispan

----------------------
   hispan |      Freq.
----------+-----------
        0 |      2,208
        1 |        218
----------------------
```

Our dependent variable is a “feeling thermometer” toward Latinos. Let’s estimate a bivariate model with the lone dummy variable:

```
. reg hispthrm hispan

      Source |       SS       df       MS              Number of obs =    2076
-------------+------------------------------           F(  1,  2074) =  179.16
       Model |  61137.8608     1  61137.8608           Prob > F      =  0.0000
    Residual |  707756.457  2074  341.251908           R-squared     =  0.0795
-------------+------------------------------           Adj R-squared =  0.0791
       Total |  768894.317  2075  370.551478           Root MSE      =  18.473

------------------------------------------------------------------------------
    hispthrm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      hispan |   19.33342   1.444412    13.38   0.000     16.50077    22.16606
       _cons |   59.43753   .4241348   140.14   0.000     58.60576    60.26931
```

The interpretation? Here we see the coefficient for the Latino indicator is positive and significantly different from 0. The value of 19.33 indicates the following: compared to non-Latino respondents, Latino respondents rate the Latino group about 19 points (or “degrees”) higher. That is, ratings are about 19 points higher than the baseline group of non-Latino respondents.

Clearly, there will only be two predicted values. For Latino respondents, y-hat is:

```
. display _b[_cons]+_b[hispan]*1
78.77095
```

and for non-Latino respondents, y-hat is:

```
. display _b[_cons]+_b[hispan]*0
59.437533
```

For non-Latinos, it’s easy to see that the predicted value of Y is given by the intercept. If we use Stata’s predict option, we could retrieve the predicted values:

```
. predict xb, xb

. table xb hispan

------------------------
Linear    |
predictio |    hispan
n         |     0      1
----------+-------------
 59.43753 | 2,208
 78.77095 |          218
```

Before moving on, let me illustrate another point. The coding of the dummy variable is arbitrary. Suppose we “flip-flop” the dummy coding and make Latino’s the “0” group and non-Latino’s the “1” group?

```
. gen hispan2=1-hispan

. reg hispthrm hispan2

      Source |       SS       df       MS              Number of obs =    2076
-------------+------------------------------           F(  1,  2074) =  179.16
       Model |  61137.8608     1  61137.8608           Prob > F      =  0.0000
    Residual |  707756.457  2074  341.251908           R-squared     =  0.0795
```

```
-------------+------------------------------           Adj R-squared =  0.0791
       Total |  768894.317  2075  370.551478           Root MSE      =  18.473

------------------------------------------------------------------------------
    hispthrm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     hispan2 |  -19.33342   1.444412   -13.38   0.000    -22.16606   -16.50077
       _cons |   78.77095   1.380737    57.05   0.000     76.06317    81.47873
------------------------------------------------------------------------------
```

What's the difference? NONE. The model produces the exact same results (as it must). The interpretation of the dummy variable's coefficient changes a bit. Now the coefficient is in reference to a different baseline. Thus, what we see is that when compared to Latinos, non-Latino respondent's ratings of the Latino group is about 19 points lower than the baseline group (subtract 19.33 from 78.77 and you'll get the intercept from the first version of this model).

Now let's illustrate one last point. To see again how arbitrary the coding of categorical variables is, consider the following coding "system." Suppose we gave Latinos the code of "151" and non-Latinos the score of "-75"? Then we apply regression:

```
. gen hispan3=151 if hispan2==1
(218 missing values generated)

. replace hispan3=-75 if hispan2==0
(218 real changes made)

. reg hispthrm hispan3

      Source |       SS       df       MS              Number of obs =    2076
-------------+------------------------------           F(  1,  2074) =  179.16
       Model |  61137.8608     1  61137.8608           Prob > F      =  0.0000
    Residual |  707756.457  2074  341.251908           R-squared     =  0.0795
-------------+------------------------------           Adj R-squared =  0.0791
       Total |  768894.317  2075  370.551478           Root MSE      =  18.473

------------------------------------------------------------------------------
    hispthrm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     hispan3 |  -.0855461   .0063912   -13.38   0.000    -.0980799   -.0730123
       _cons |   72.35499   .9332037    77.53   0.000     70.52488    74.18511
------------------------------------------------------------------------------
```

The results look different (but note variance components), but they're not. To see, look at the predicted values:

```
. predict xbalt, xb

. table xb

----------------------
Linear    |
predictio |
n         |      Freq.
----------+-----------
 59.43753 |      2,208
 78.77095 |        218
----------------------
```

The y-hats are, as they must be, identical.

Note that we could plot our y-hats, but it would not make much sense to do that. Since there are only two predicted values, there would be only two "dots" in our graph.

PART II: Some Motivation

To motivate the consideration of dummy variables, let's look at the problem "backwards" for a minute. Consider the following data set (which is simulated data):

```
. list y x1 d

             y         x1          d
  1.       100         20          1
  2.       105         30          1
  3.       114         40          1
  4.       132         50          1
  5.       134         60          1
  6.        76          0          0
  7.        82          2          0
  8.        90         10          0
  9.        98         18          0
 10.       103         28          0
```

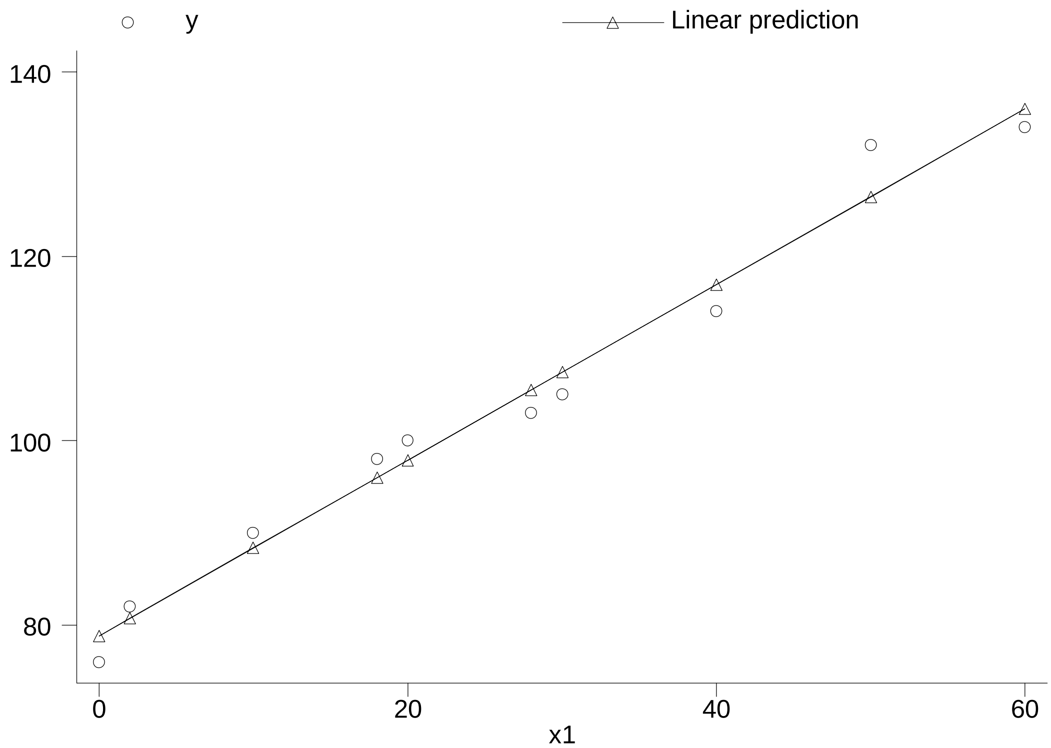
Here, y is our dependent variable, x1 is an independent variable, and d is a categorical, or dummy variable, denoting group ``1'' (d=1) and group ``2'' (d=0).

Suppose we run a regression of y on x1?

```
. reg y x1

      Source |       SS       df       MS              Number of obs =      10
-------------+------------------------------           F(  1,     8) =  338.31
       Model |   3222.2047     1   3222.2047           Prob > F      =  0.0000
    Residual |  76.1952976     8  9.52441219           R-squared     =  0.9769
-------------+------------------------------           Adj R-squared =  0.9740
       Total |     3298.40     9  366.488889           Root MSE      =  3.0862

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .9519631   .0517562    18.39   0.000      .832613    1.071313
       _cons |   78.83935   1.653934    47.67   0.000     75.02537    82.65333
------------------------------------------------------------------------------
```

We see x1 is significantly related to y. A unit increase in x1 results in about .95 change to the expected value of Y. The *t*-ratio is highly significant as well. Now, let us graph y and x1 along with the predicted values:

```
. predict xb, xb

. gr y xb x1, ylab xlab c(.l.)
```



As the model suggests, the predicted regression function closely fits the observed data. Here, drawing conclusions about y based on x1 is "not a bad thing to do."

But what about differences across the two groups? This model does not "control" for these differences? Should we? Would we learn more if we accounted for "group differences?" Suppose we estimated two submodels, one for each group? [This may seem like the right thing to do in applied work.]

The model for group 1 (d=1) is:

```
. reg y x1 if d==1

      Source |       SS       df       MS              Number of obs =       5
-------------+------------------------------           F(  1,     3) =   50.61
       Model |      902.50     1      902.50           Prob > F      =  0.0057
    Residual |       53.50     3  17.8333333           R-squared     =  0.9440
```

```
-------------+------------------------------           Adj R-squared =  0.9254
       Total |      956.00     4      239.00           Root MSE      =   4.223

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |        .95   .1335415     7.11   0.006     .5250113    1.374989
       _cons |         79   5.665686    13.94   0.001     60.96926    97.03074
```

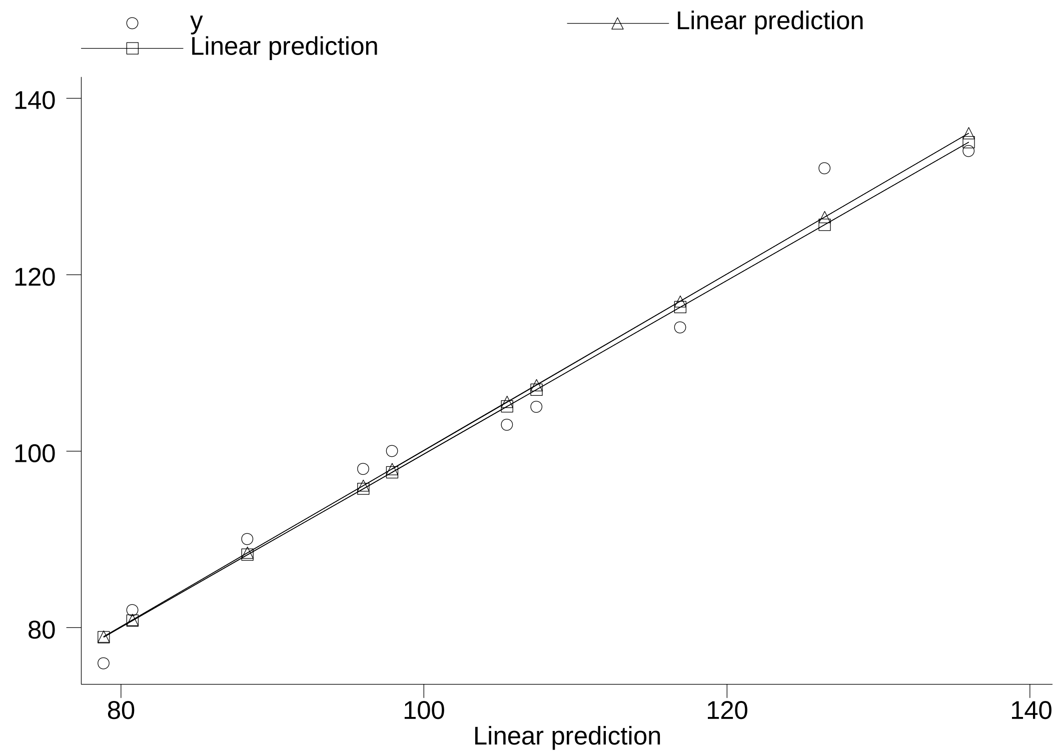
The model for group 2 (d=0) is:

```
. reg y x1 if d==0

      Source |       SS       df       MS              Number of obs =       5
-------------+------------------------------           F(  1,     3) =   62.85
       Model |  470.350445     1  470.350445           Prob > F      =  0.0042
    Residual |  22.4495549     3  7.48318497           R-squared     =  0.9544
-------------+------------------------------           Adj R-squared =  0.9393
       Total |      492.80     4      123.20           Root MSE      =  2.7355

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .9339763   .1178062     7.93   0.004     .5590642    1.308888
       _cons |   78.96588   1.834149    43.05   0.000     73.12879    84.80296
```

For either model, the interpretation is standard. If we stopped here, we would conclude that x1 is significantly related to y for both groups. But does it make sense "to control" for group affiliation in this context? The answer is no.

To see why, consider the following graph. Here we plot the predicted values from the two submodels (just estimated above):

```
. gr y xb1 xb2 xb, c(.ll.) ylab xlab
```



The two slopes are almost identical. Group 1's "slope" is the top line and Group 2's "slope" is the bottom line. There *is* a difference between the two groups but the difference is negligible in this instance. Given the similarity in slopes across the two groups, should we expect there to be significant differences between the groups?

We could directly test this by incorporating our dummy variable into the model:

```
. reg y x1 d

      Source |       SS       df       MS              Number of obs =      10
-------------+------------------------------           F(  2,     7) =  148.32
       Model |  3222.3605     2  1611.18025            Prob > F      =  0.0000
    Residual |  76.039501     7  10.8627859            R-squared     =  0.9769
-------------+------------------------------           Adj R-squared =  0.9704
       Total |    3298.40     9  366.488888            Root MSE      =  3.2959

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.       t     P>|t|      [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .9443867   .0840085     11.24   0.000      .7457382    1.143035
           d |   .3794179   3.168178      0.12   0.908     -7.112132    7.870968
       _cons |   78.84511   1.766976     44.62   0.000      74.66688    83.02335
```

What do we find? We see that "controlling" for *d* makes no difference in our conclusions about the relationship between y and x1. The relationship between these two variables is almost identical to a model not including the dummy variable.

This illustration demonstrates what it means for a dummy to be insignificant. In the context of this model, there are no appreciable group differences when we account for x1. Interestingly, if we ignored x1 and simply ran a dummy variable only model, we would obtain:

```
. reg y d

      Source |       SS       df       MS              Number of obs =      10
-------------+------------------------------           F(  1,     8) =   10.21
       Model |    1849.60     1     1849.60            Prob > F      =  0.0127
    Residual |    1448.80     8      181.10            R-squared     =  0.5608
-------------+------------------------------           Adj R-squared =  0.5059
       Total |    3298.40     9  366.488889            Root MSE      =  13.457

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.       t     P>|t|      [95% Conf. Interval]
-------------+----------------------------------------------------------------
           d |       27.2   8.511169      3.20   0.013      7.573209    46.82679
       _cons |       89.8   6.018305     14.92   0.000      75.92176    103.6782
------------------------------------------------------------------------------
```

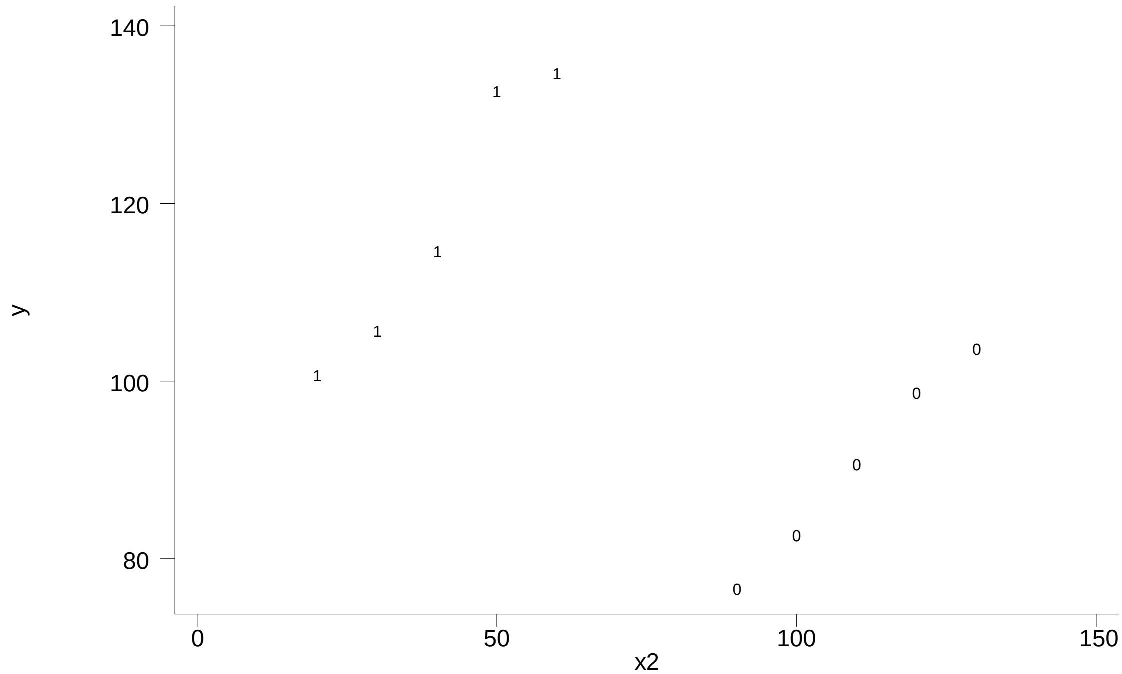
This model seems to suggest substantial group differences; however, once we account for x1, these group differences are explained (in terms of x1) and so the apparently large effect of d goes away.

Now let's take a different tack. Consider the following data:

```
. list y x2 d
             y        x2          d
  1.       100        20          1
  2.       105        30          1
  3.       114        40          1
  4.       132        50          1
  5.       134        60          1
  6.        76        90          0
  7.        82       100          0
  8.        90       110          0
  9.        98       120          0
 10.       103       130          0
```

Here, y and d are as before and x2 is a quantitative variable. Let's graph y and x2:

```
gr y x2, s([d]) ylab xlab
```



In this graph, I've used the code for the variable "d" to denote where the observations for the dummy coding lie. In the graph, happily, we see that both groups seem to display a positive relationship between x2 and y: as x2 increases, y seems to increase for both groups. Indeed, the groupwise correlation coefficients are nearly unity:

```
. corr y x2 if d==1
(obs=5)

             |        y       x2
-------------+------------------
           y |   1.0000
          x2 |   0.9716   1.0000

. corr y x2 if d==0
(obs=5)

             |        y       x2
-------------+------------------
           y |   1.0000
          x2 |   0.9972   1.0000
```

Knowing this, suppose we estimate a bivariate regression model for *each group separately*. Thus, for d=1, we would get:

```
. reg y x2 if d==1

      Source |       SS       df       MS              Number of obs =       5
-------------+------------------------------           F(  1,     3) =   50.61
       Model |      902.50     1      902.50           Prob > F      =  0.0057
    Residual |       53.50     3  17.8333333           R-squared     =  0.9440
-------------+------------------------------           Adj R-squared =  0.9254
       Total |      956.00     4      239.00           Root MSE      =   4.223

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x2 |        .95   .1335415     7.11   0.006     .5250113    1.374989
       _cons |         79   5.665686    13.94   0.001     60.96926    97.03074
------------------------------------------------------------------------------
```

For d=0 we would get:

```
. reg y x2 if d==0

      Source |       SS       df       MS              Number of obs =       5
-------------+------------------------------           F(  1,     3) =  525.00
       Model |      490.00     1      490.00           Prob > F      =  0.0002
    Residual |        2.80     3  .933333333           R-squared     =  0.9943
-------------+------------------------------           Adj R-squared =  0.9924
       Total |      492.80     4      123.20           Root MSE      =  .96609

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x2 |         .7   .0305505    22.91   0.000     .6027747    .7972253
       _cons |       12.8   3.388215     3.78   0.032     2.017188    23.58281
------------------------------------------------------------------------------
```

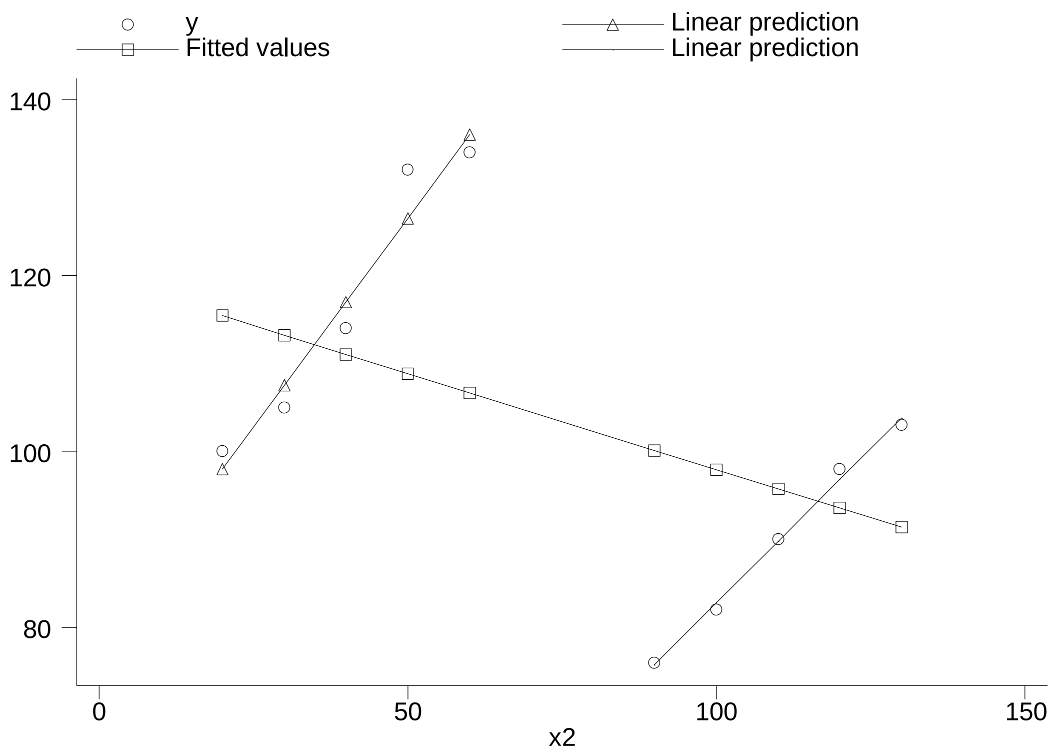
Our two submodels say x2 is strongly related to y for each group. Thus, we may have confidence that in our full sample, x2 is strongly related to y. Now, let's estimate the model:

```
. reg y x2

      Source |       SS       df       MS              Number of obs =      10
-------------+------------------------------           F(  1,     8) =    2.07
       Model |   678.74386     1   678.74386           Prob > F      =  0.1879
    Residual |  2619.65614     8  327.457018           R-squared     =  0.2058
-------------+------------------------------           Adj R-squared =  0.1065
       Total |     3298.40     9  366.488889           Root MSE      =  18.096

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x2 |  -.2182456   .1515897    -1.44   0.188    -.5678121    .1313209
       _cons |   119.7684   12.72812     9.41   0.000     90.41732    149.1195
```

What???

When we combine the groups, we find that the regression coefficient for x2 is now *negative* whereas before, the coefficients for each group separately were strongly *positive*. How can this be?

Consider the following graph, which plots the predicted values from the three previous models:



What do we find? The OLS estimator for the full sample produces a downwardly sloping line (i.e. the residuals are minimized in this way?) even though the two submodels yield strongly positive slopes.

This is an example of SIMPSON'S PARADOX. When data from two or more groups are combined, if the direction of the relationship changes in the combined data set, then this is Simpson's paradox. More succinctly, as Fox puts it: "that marginal and partial relationships can differ in sign is called "Simpson's Paradox." (p. 136). The marginal relationship between x2 and y is negative but the partial relationship (controlling for d) is positive.

In the "full model" we omitted the variable d yet we know that the effect of x2 on y is related to d. For both groups the effect is positive but since the mean level of x2 is much, much higher for d=0, omitting d gives us the false conclusion that x2 and y are negatively related.

One way (not the only) to avoid this problem (and it can be a *real problem* with small *n* analyses) is to account for the omitted variable, which in the previous model is d.

```
. reg y x2 d

      Source |       SS       df       MS              Number of obs =      10
-------------+------------------------------           F(  2,     7) =  128.36
       Model |     3210.85     2    1605.425           Prob > F      =  0.0000
    Residual |       87.55     7  12.5071429           R-squared     =  0.9735
-------------+------------------------------           Adj R-squared =  0.9659
       Total |     3298.40     9  366.488889           Root MSE      =  3.5365

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x2 |       .825   .0790795    10.43   0.000     .6380066    1.011993
           d |      84.95   5.970373    14.23   0.000     70.83231    99.06769
       _cons |       -.95    8.84136    -0.11   0.917    -21.85649    19.95649
```

Now look what we have. After accounting for d (or "controlling" for it as this is sometimes called), the positive relationship between x2 and y is retrieved.
As noted in lecture notes, this model will produce "parallel" slopes, which is equivalent to saying, there will be two regression functions, one for d=1, one for d=0. To illustrate, the predicted value of y when d=1 is given by:

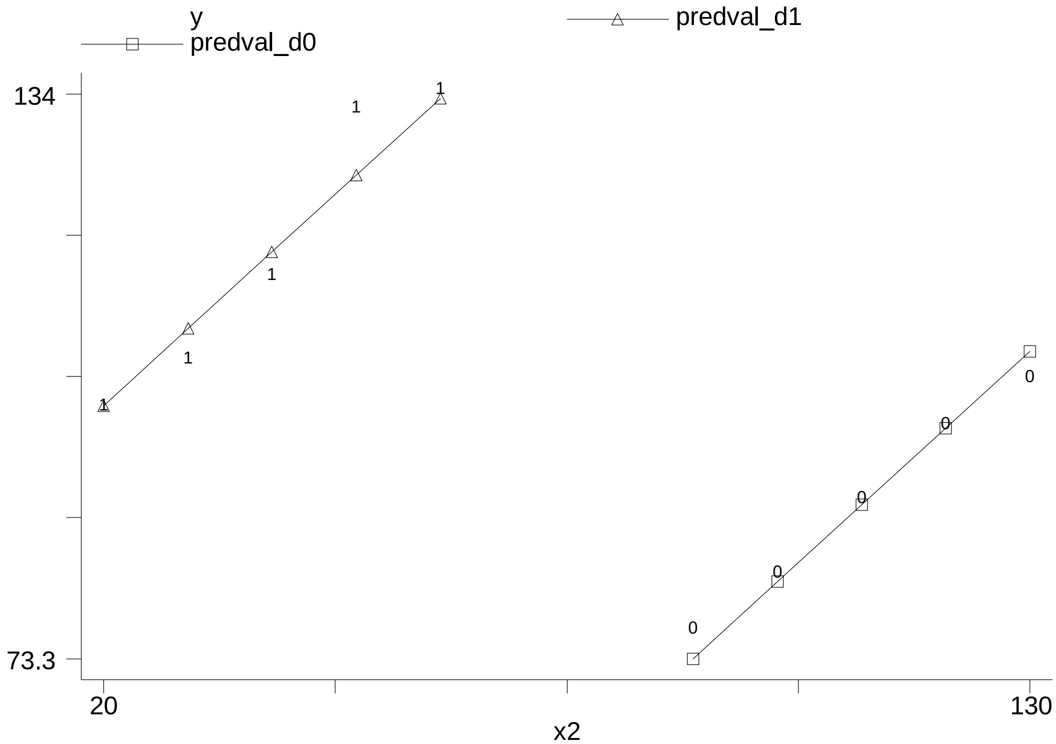
```
. gen predval_d1=(_b[_cons]+_b[d])+_b[x2]*x2   if d==1
(5 missing values generated)
```

and the predicted value of y when d=0 is given by:

```
. gen predval_d0=_b[_cons]+_b[x2]*x2   if d==0
(5 missing values generated)
```

(Verify that these statements correspond to functions from lecture notes).

Graphing the predicted values (which could have been generated by Stata's predict option) gives us:



Now the regression function looks much better (i.e. conforms to our expectations) after accounting for d than by omitting d. Simpson's Paradox is avoided.

However, we now have another potential problem to worry about: the assumption of parallel slopes. In the context of this model, we're saying that the difference between the two groups is solely attributable to differences in intercepts. Here, the two groups are offset *d* but the slopes for x2 are identical across the two groups. This is the parallel slopes assumption.

PART III: Polytomous Variables

Suppose we have a variable that is multicategorical? Let's return to the evaluation of Latinos model.

```
. table raceeth

----------------------
  raceeth |      Freq.
----------+-----------
        0 |      1,896
        1 |        312
        2 |        218
----------------------
```

Here, a "0" denotes white, non-Latinos; a "1" denotes African-American respondents; and a "2" denotes Latino respondents. It is categorical (we could flip the coding around and it wouldn't matter), but is it ordered? Is category 2 twice that of the baseline category? Obviously, the answer is no. There is no ordinality here; these are clearly categorical data. Suppose we run a regression despite this (which is *commonly done by beginning data analysts*)?

```
. reg hispthrm raceeth

      Source |       SS       df       MS              Number of obs =    2076
-------------+------------------------------           F(  1,  2074) =  238.48
       Model |  79293.3793     1  79293.3793           Prob > F      =  0.0000
    Residual |  689600.938  2074  332.498042           R-squared     =  0.1031
-------------+------------------------------           Adj R-squared =  0.1027
       Total |  768894.317  2075  370.551478           Root MSE      =  18.235

------------------------------------------------------------------------------
    hispthrm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     raceeth |   10.01341   .6484228    15.44   0.000     8.741787    11.28504
       _cons |   58.13813   .4439162   130.97   0.000     57.26756    59.00869
------------------------------------------------------------------------------
```

We get a significant coefficient but what does it mean? Look at the predicted values:

```
. table xb

----------------------
Fitted    |
values    |      Freq.
----------+-----------
 58.13813 |      1,896
 68.15154 |        312
 78.16496 |        218
----------------------
```

Are we really to believe that differences across race/ethnic groups is a constant 10.01? The OLS model above treats X as ordinal and further, treats the distance between a 0 and 1 and a 1 and a 2 equivalently. It assumes you "know" the distance between adjacent categories, but of course, you don't.

The solution: partition X into separate dummy variables:

Distribution of dummy variables:

```
. table white2

----------------------
   white2 |      Freq.
----------+-----------
        0 |        563
        1 |      1,896
----------------------

. table black

----------------------
    black |      Freq.
----------+-----------
        0 |      2,147
        1 |        312
----------------------

. table hispan

----------------------
Ind. Lev. |
Hispanic  |      Freq.
----------+-----------
        0 |      2,267
        1 |        218
----------------------
```

Now we have three dummies. Suppose we estimate a regression:

```
reg hispthrm white2 black hispan

      Source |       SS       df       MS              Number of obs =    2058
-------------+------------------------------           F(  2,  2055) =  105.34
       Model |  69786.5003     2  34893.2501           Prob > F      =  0.0000
    Residual |  680723.982  2055  331.252546           R-squared     =  0.0930
-------------+------------------------------           Adj R-squared =  0.0921
       Total |  750510.482  2057  364.856822           Root MSE      =   18.20

------------------------------------------------------------------------------
```

```
    hispthrm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      white2 |  -19.83921   1.503148   -13.20   0.000    -22.78706   -16.89136
       black |  -10.73609   1.829314    -5.87   0.000    -14.32359   -7.148583
      hispan |  (dropped)
       _cons |   78.04348   1.434388    54.41   0.000     75.23047    80.85648
```

OOPS. What happened?

Why did the "hispan" variable drop out?

Let's reestimate but omit one of the dummies:

```
reg hispthrm black hispan if insample==1

      Source |       SS       df       MS              Number of obs =    2058
-------------+------------------------------           F(  2,  2055) =  105.34
       Model |  69786.5003     2  34893.2501           Prob > F      =  0.0000
    Residual |  680723.982  2055  331.252546           R-squared     =  0.0930
-------------+------------------------------           Adj R-squared =  0.0921
       Total |  750510.482  2057  364.856822           Root MSE      =   18.20

------------------------------------------------------------------------------
    hispthrm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       black |   9.103125   1.221026     7.46   0.000     6.708548     11.4977
      hispan |   19.83921   1.503148    13.20   0.000     16.89136    22.78706
       _cons |   58.20427   .4494255   129.51   0.000     57.32289    59.08565
```

What Happened?

We have estimates now. Why? What does the constant term represent?

Generate Predicted Values:

```
. predict xb1
(option xb assumed; fitted values)
(26 missing values generated)

. table xb1

----------------------
Fitted    |
values    |      Freq.
----------+-----------
 58.20427 |      1,951
  67.3074 |        312
 78.04348 |        196
```

Why 3 predicted values?

Y-hat for whites: 58
Y-hat for blacks: 67
Y-hat for Hispanics: 78

How do I know this? "Unpack" the model and see for yourself. Where do the y-hats come from? Using multiple dummies is not particularly different from a single dummy variable. The model must be broken down into submodels in order to derive predicted effects. This is exactly what we did before.

With multiple dummy variables, the choice of dummy to omit is arbitrary.

Don't believe me? Try this model:

```
. reg hispthrm white2 hispan if insample==1

      Source |       SS       df       MS              Number of obs =    2058
-------------+------------------------------           F(  2,  2055) =  105.34
       Model |  69786.5003     2  34893.2501           Prob > F      =  0.0000
    Residual |  680723.982  2055  331.252546           R-squared     =  0.0930
-------------+------------------------------           Adj R-squared =  0.0921
       Total |  750510.482  2057  364.856822           Root MSE      =   18.20

------------------------------------------------------------------------------
    hispthrm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      white2 |  -9.103125   1.221026    -7.46   0.000     -11.4977   -6.708548
      hispan |   10.73609   1.829314     5.87   0.000     7.148583    14.32359
```

```
       _cons |   67.30739   1.135306    59.29   0.000     65.08092    69.53386
------------------------------------------------------------------------------

. predict xb2
(option xb assumed; fitted values)
(26 missing values generated)

. table xb2

----------------------
Fitted    |
values    |      Freq.
----------+-----------
 58.20427 |      1,896
  67.3074 |        367
 78.04348 |        196
----------------------
```

What's the difference? Black category is omitted.

Implications for results? NONE. See predicted values.

Coding in this way is not the only method of coding dummy variables. Consider "contrast coding."

Illustration of contrast coding.

Here we recode the dummy variables in a way that the regression coefficients tell us what the differences are across levels of race. The comparison group changes. In the previous model, the comparison was always against the baseline category.

```
. gen blackhisp=1 if black==1 | hispan==1
(1955 missing values generated)

. replace blackhisp=0 if blackhisp==.
(1955 real changes made)

. gen hisp=1 if hispan==1
(2267 missing values generated)

. replace hisp=0 if hisp==.
(2267 real changes made)

. table blackhisp black hispan

----------------------------------------
          | Ind. Lev. Hispanic and black
          | ----- 0 ----    ----- 1 ----
blackhisp |     0      1        0      1
----------+-----------------------------
        0 | 1,951
        1 |          312      196
----------------------------------------

. table hisp  hispan

------------------------
          |  Ind. Lev.
          |   Hispanic
     hisp |     0      1
----------+-------------
        0 | 2,267
        1 |          218
------------------------
```

Now I estimate the regression model:

```
. reg hispthrm blackhisp hisp if insample==1

      Source |       SS       df       MS              Number of obs =    2076
-------------+------------------------------           F(  2,  2073) =  119.61
       Model |  79549.4224     2  39774.7112           Prob > F      =  0.0000
    Residual |  689344.895  2073  332.534923           R-squared     =  0.1035
-------------+------------------------------           Adj R-squared =  0.1026
       Total |  768894.317  2075  370.551478           Root MSE      =  18.236

------------------------------------------------------------------------------
    hispthrm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
```

```
-------------+----------------------------------------------------------------
   blackhisp |   9.103125   1.223387     7.44   0.000      6.70393    11.50232
        hisp |   11.46356   1.775288     6.46   0.000     7.982024    14.94509
       _cons |   58.20427   .4502946   129.26   0.000     57.32119    59.08735
------------------------------------------------------------------------------

. predict xb3
(option xb assumed; fitted values)

. table xb3

----------------------
Fitted    |
values    |      Freq.
----------+-----------
 58.20427 |      1,955
  67.3074 |        312
 78.77095 |        218
```

What's the difference? In terms of predicted values, there are none; however, in this model, the coefficients tell you how the difference between African Americans vs. whites (the blackhisp coefficient) and the difference between Latinos vs. African Americans (the Hispanic coefficient). In the other models, the reference category was always the baseline.

Hence, the model is unpacked in the following way:

y-hat=b0 for whites;
y-hat=b0 + b1(BLACKHISP) for blacks;
y-hat=b0+ b1(BLACKHISP) + b2(HISP) for Latinos.

We see that Latino evaluations are 11 points higher than blacks; blacks evaluation are 9 points higher than whites. Finally, we see that Latino evaluations are about 20 points higher than whites. This is a useful alternative to the usual way to code dummy variables. NOTE: it has no implications for predicted values!

The issues discussed previously with parallel slopes hold here as well. That is, with a quantitative variable, the model would produce parallel slopes.

Suppose we have X1, and it's a quantitative variable. Consider the model:

```
. reg hispthrm white2 hispan egal if insample==1

      Source |       SS       df       MS              Number of obs =    2020
-------------+------------------------------           F(  3,  2016) =   90.26
       Model |  86914.134     3  28971.378             Prob > F      =  0.0000
    Residual |  647097.22  2016  320.980764            R-squared     =  0.1184
-------------+------------------------------           Adj R-squared =  0.1171
       Total | 734011.354  2019  363.551934            Root MSE      =  17.916

------------------------------------------------------------------------------
    hispthrm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      white2 |  -6.224735    1.27034    -4.90   0.000    -8.716051   -3.733418
      hispan |   12.50433   1.838494     6.80   0.000     8.898787    16.10988
        egal |   .6862087   .0867336     7.91   0.000     .5161118    .8563056
       _cons |   54.99232   1.922148    28.61   0.000     51.22272    58.76192
------------------------------------------------------------------------------
```

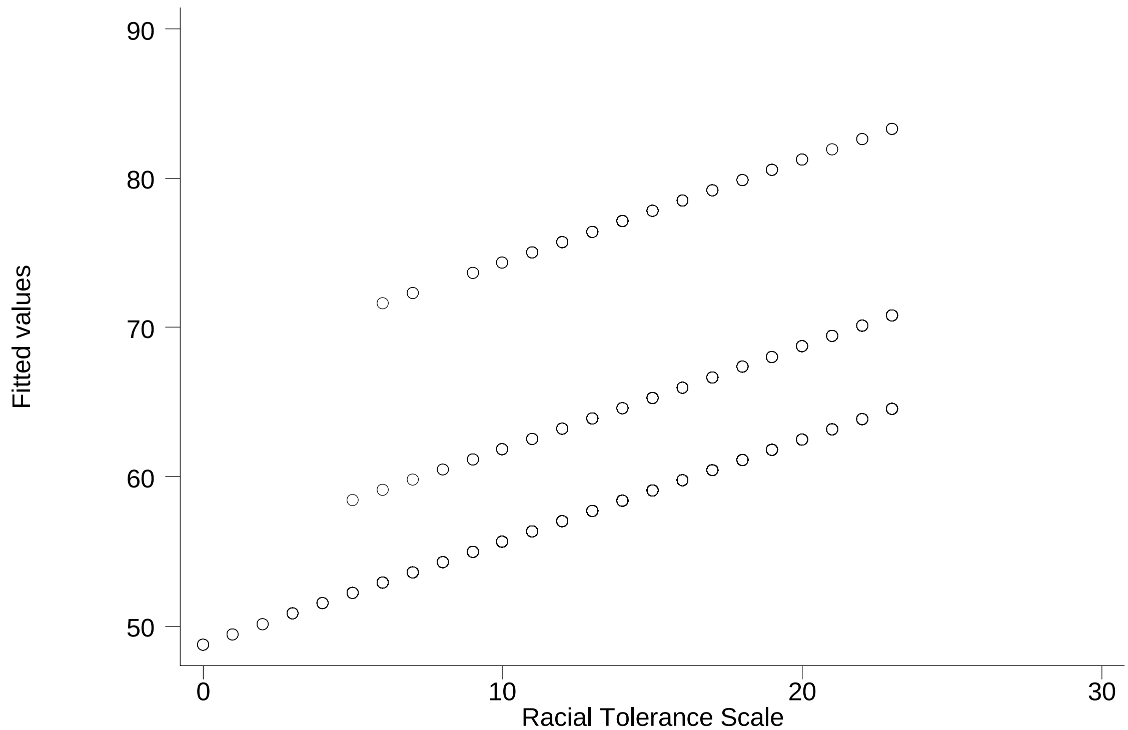
Here we have our dummy variables model with the quantitative covariate. It is significantly different from 0.

What the predicted values?

```
. predict xb4
(option xb assumed; fitted values)
(319 missing values generated)
```

Graph them:

```
gr xb4 egal, ylab xlab b2("Racial Tolerance Scale")
```



What is the central feature of this graph? What are the implications here?

What is the alternative?

NONPARALLEL SLOPES