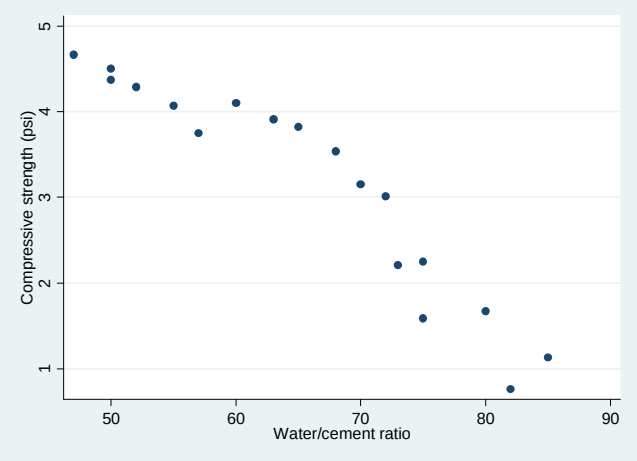
MA398 – Brief Overview of Piecewise Linear Regression, Weighted Least Squares, and Logistic Regression

Piecewise Linear Regression

For the first part of this exercise we will be using the *cement.dta* dataset found on the course webpage at http://www.colby.edu/personal/l/lobrien/ma397.html.

The data consist of compressive strength (the response) and water/cement ratio (the predictor) for 18 batches of cement. A plot of the data is below:



Notice that there is not a single linear relationship. One could argue that there is a quadratic relationship here, so let's fit a second-order model:

```
. regress  strength ratio ratiosq

      Source |       SS       df       MS              Number of obs =      18
-------------+------------------------------           F(  2,    15) =  106.54
       Model |  24.6055287     2  12.3027644           Prob > F      =  0.0000
    Residual |   1.7321213    15  .115474753           R-squared     =  0.9342
-------------+------------------------------           Adj R-squared =  0.9255
       Total |    26.33765    17  1.54927353           Root MSE      =  .33982

------------------------------------------------------------------------------
    strength |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       ratio |   .1973654   .0890095     2.22   0.042     .0076462    .3870845
     ratiosq |  -.0022655   .0006789    -3.34   0.005    -.0037126   -.0008183
       _cons |   .2480465   2.846975     0.09   0.932    -5.820138    6.316231
------------------------------------------------------------------------------
```

We see that the quadratic term is significant indicating the presence of a quadratic trend. However, we may also fit a piecewise linear regression to these data if we consider the data for ratios less than 70 to have one linear relationship, and data above 70 to have a different linear relationship. To do this, we need to generate a knot indicator that tells us which piece of the data we are in:

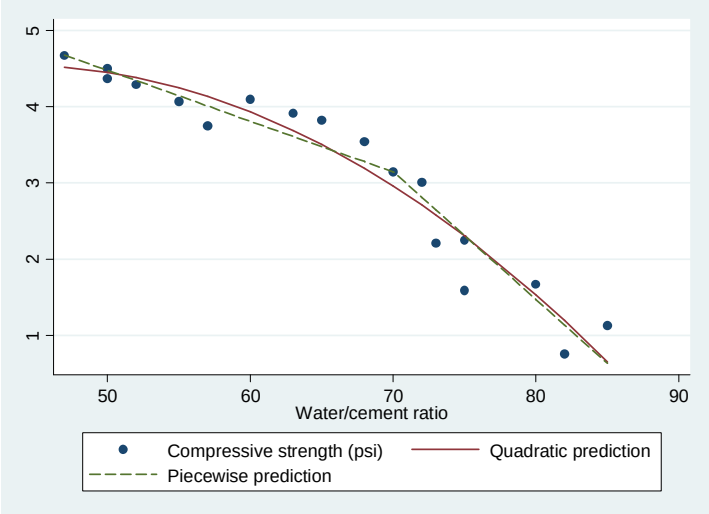
```
. gen x=1 if ratio > 70
(11 missing values generated)

. replace x=0 if ratio <= 70
(11 real changes made)
```

Now generate a variable that multiplies this indicator by the difference between the observed ratio and knot at 70):

```
. gen x2 = x*(ratio-70)
```

The regression model we wish to run therefore is $E(y) = \beta_0 + \beta_1(ratio) + \beta_2 x_2$. We obtain:

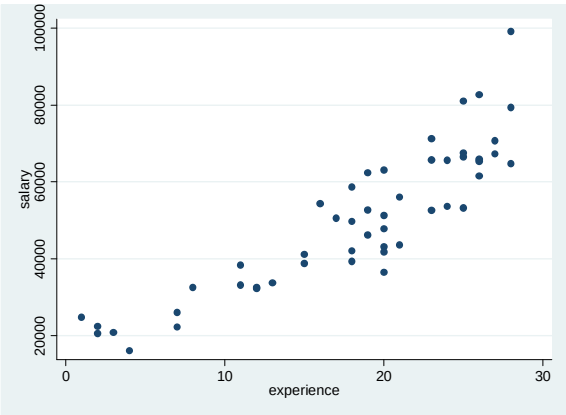
```
. regress strength ratio x2

      Source |       SS       df       MS              Number of obs =      18
-------------+------------------------------           F(  2,    15) =  114.44
       Model |  24.7177524     2  12.3588762           Prob > F      =  0.0000
    Residual |   1.6198976    15  .107993173           R-squared     =  0.9385
-------------+------------------------------           Adj R-squared =  0.9303
       Total |    26.33765    17  1.54927353           Root MSE      =  .32862

------------------------------------------------------------------------------
    strength |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       ratio |  -.0663308   .0112348    -5.90   0.000    -.0902771   -.0423845
          x2 |  -.1011861   .0281245    -3.60   0.003     -.161132   -.0412402
       _cons |   7.791983   .6769605    11.51   0.000     6.349076     9.23489
------------------------------------------------------------------------------
```



Which model fits the data better? It seems that the piecewise model is slightly better but not by much. Either would probably suffice in this example. However there are times where the piecewise models will outperform all other deterministic models.

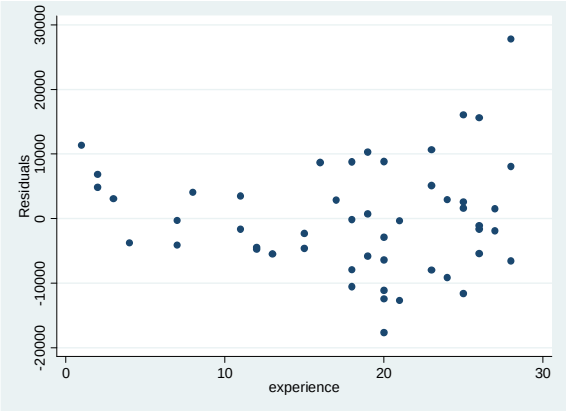
Weighted Least Squares

Consider the dataset *socwork.dta* on the course webpage. Recall that it contains information on social workers' salaries (y) and their years of experience (x). The data are plotted below.



If we run the OLS regression we obtain the following output and residual plot.

```
. regress  salary experience

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  1,    48) =  177.26
       Model |  1.3240e+10     1  1.3240e+10           Prob > F      =  0.0000
    Residual |  3.5852e+09    48  74691793.3           R-squared     =  0.7869
-------------+------------------------------           Adj R-squared =  0.7825
       Total |  1.6825e+10    49   343364521           Root MSE      =  8642.4

------------------------------------------------------------------------------
      salary |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  experience |   2141.381   160.8392    13.31   0.000     1817.992     2464.77
       _cons |   11368.72   3160.317     3.60   0.001     5014.478    17722.96
------------------------------------------------------------------------------
```

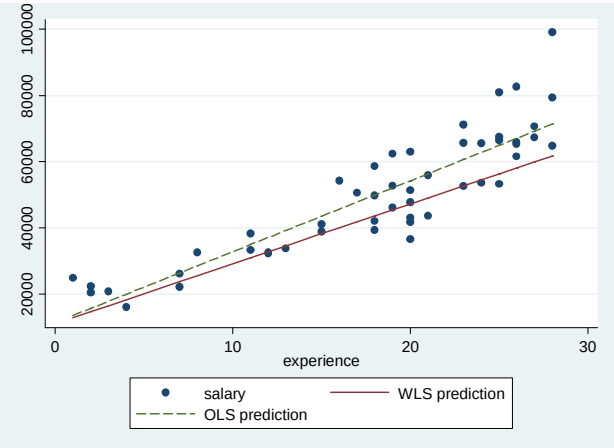


You can see that the assumption of homoscedasticity has been violated here. Weighted least squares may help to alleviate this problem.

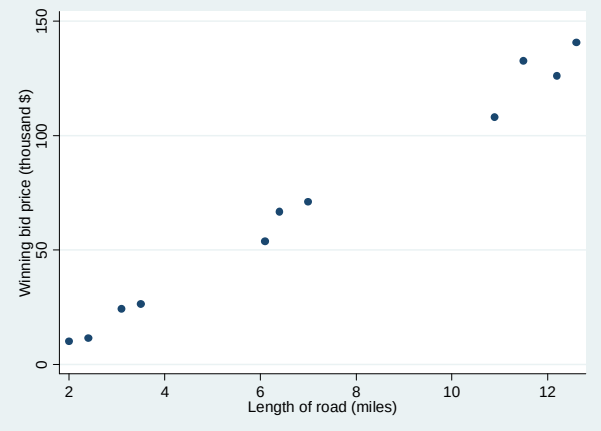
Stata perform WLS estimation in two different ways. You may either provide a variable that contains an estimate of the conditional standard deviation of y given x, or you may let Stata treat the predictor as a categorical variable and have it determine the weights automatically by taking the inverse of the variance of the y's at each value of x. Since "experience" in the *socwork.dta* data is a categorical variable 9albeit quantitative), we can do the latter. To run a WLS regression of this type in Stata give the command:

```
. vwls  salary experience

Variance-weighted least-squares regression          Number of obs   =      43
Goodness-of-fit chi2(13)   =  110.97               Model chi2(1)   =  771.22
Prob > chi2                =  0.0000               Prob > chi2     =  0.0000
------------------------------------------------------------------------------
      salary |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  experience |   1810.416   65.19111    27.77   0.000     1682.643    1938.188
       _cons |   11018.06   792.4572    13.90   0.000      9464.87    12571.24
------------------------------------------------------------------------------
```

The difference between the OLS and WLS estimates can be seen below:

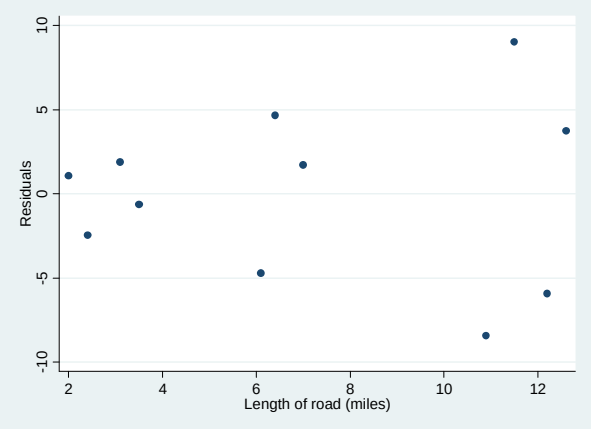


To illustrate the use of the procedure outlined in the book consider the dataset *dot11.dta*. This contains data on the length of road (x) and the winning bid price (y) for 11 construction jobs. The plot is below.



The OLS regression yields the following output and residual plot.

```
. regress price length

      Source |       SS       df       MS              Number of obs =      11
-------------+------------------------------           F(  1,     9) =  850.45
       Model |  24557.8778     1  24557.8778           Prob > F      =  0.0000
    Residual |   259.88646     9  28.8762733           R-squared     =  0.9895
-------------+------------------------------           Adj R-squared =  0.9884
       Total |  24817.7642    10  2481.77642           Root MSE      =  5.3737

------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      length |   12.06868   .4138422    29.16   0.000      11.1325    13.00485
       _cons |  -15.11237   3.342214    -4.52   0.001    -22.67298   -7.551758
------------------------------------------------------------------------------
```



You can see that there is a violation of homoscedasticity, and that the variability of the residuals seems to different in three distinct regions. We will calculate that residual variance in each of the three regions (2,4,), (6,7), and (10,13) and relate those variances to functions of the mean of length (x).

```
. gen cat=1 if length >=2 & length <=4
(7 missing values generated)

. replace cat=2 if length >=6 & length <=7
(3 real changes made)

. replace cat=3 if length >=10 & length <=13
(4 real changes made)

. predict resid, r

. sort cat

. by cat: summ length resid

-------------------------------------------------------------------------------------------------------
-> cat = 1

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      length |         4        2.75    .6757711          2        3.5
       resid |         4   -.0264858    1.929575  -2.452451    1.89948
```

```
-------------------------------------------------------------------------------------------------
-------------------
-> cat = 2

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      length |         3         6.5    .4582576         6.1          7
       resid |         3    .5659814    4.797119  -4.706548   4.672851

-------------------------------------------------------------------------------------------------
-------------------
-> cat = 3

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      length |         4        11.8    .7527729        10.9       12.6
       resid |         4   -.3980002    8.187278  -8.436186   9.022601
```

Now that we have the variance of the residuals and the mean of length in each region, we can generate comparisons of these to functions of the mean of length. First we have to enter the mean (length) and variance (of the residuals) for each of the three regions.

```
. gen xbar=2.75 if length >=2 & length <=4
(7 missing values generated)

. replace xbar=6.5 if length >=6 & length <=7
(3 real changes made)

. replace xbar=11.8 if length >=10 & length <=13
(4 real changes made)

. gen vresid=1.929575^2 if length >=2 & length <=4
(7 missing values generated)

. replace vresid=4.797119^2 if length >=6 & length <=7
(3 real changes made)

. replace vresid=8.187278^2 if length >=10 & length <=13
(4 real changes made)

. gen f1=vresid/xbar

. gen f2=vresid/xbar^2

. gen f3=vresid/sqrt(xbar)
```

Now list the new functions of xbar and the variance of the residuals.

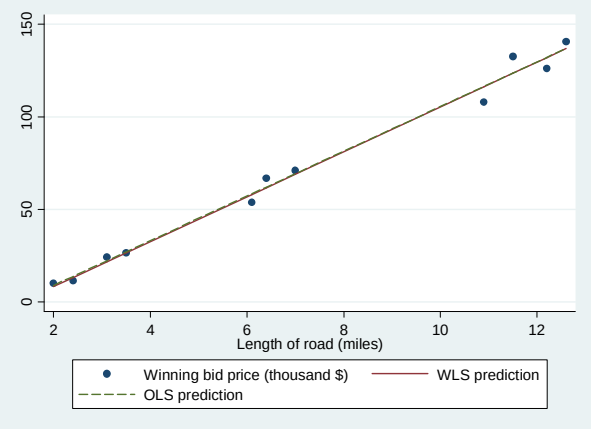
```
. list cat f1 f2 f3

     +-----------------------------------+
     | cat         f1         f2         f3 |
     |-----------------------------------|
  1. |   1   1.353913   .4923319    2.24521 |
  2. |   1   1.353913   .4923319    2.24521 |
  3. |   1   1.353913   .4923319    2.24521 |
  4. |   1   1.353913   .4923319    2.24521 |
  5. |   2   3.540362    .544671   9.026186 |
     |-----------------------------------|
  6. |   2   3.540362    .544671   9.026186 |
  7. |   2   3.540362    .544671   9.026186 |
  8. |   3   5.680637     .48141   19.51363 |
  9. |   3   5.680637     .48141   19.51363 |
 10. |   3   5.680637     .48141   19.51363 |
     |-----------------------------------|
 11. |   3   5.680637     .48141   19.51363 |
     +-----------------------------------+
```

We can see that “f2” provides values across the three regions that have values that are the most similar. This indicates that f2 provides a reasonably good function by which we can generate the weights. Now we generate those weights by defining a new variable that contains the square root of the weights:

```
. gen sqrtweight = sqrt(1/xbar^2)
```

Now we can run the WLS regression using that weighting variable:

```
. vwls  price length, sd( sqrtweight)

Variance-weighted least-squares regression          Number of obs   =      11
Goodness-of-fit chi2(9)    = 30109.41               Model chi2(1)   =  7.4e+05
Prob > chi2                =  0.0000                Prob > chi2     =  0.0000
------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      length |   12.12675   .0140712   861.81   0.000     12.09917    12.15433
       _cons |  -15.93188   .1520816  -104.76   0.000    -16.22996   -15.63381
------------------------------------------------------------------------------
```



We can see that the two regression lines using OLS and WLS are very similar in this case. However the weight LS residuals would be smaller than the OLS residuals.

Logistic Regression

Stata has a wide variety of logistic regression functions. The basic function that we will explore is the *logit* command. We will use a dataset that contains the bid status (1=fixed or 0=competitive) as the response, and the number of bidder and the differences between the winning bid value and estimated winning bid as predictors. To run this model, we issue the command:

```
. logit status bidders difference

Iteration 0:   log likelihood = -20.690383
Iteration 1:   log likelihood = -13.065427
Iteration 2:   log likelihood = -11.733602
Iteration 3:   log likelihood = -11.441967
```

```
Iteration 4:   log likelihood =  -11.42164
Iteration 5:   log likelihood = -11.421512

Logistic regression                                     Number of obs   =         31
                                                        LR chi2(2)      =      18.54
                                                        Prob > chi2     =     0.0001
Log likelihood = -11.421512                             Pseudo R2       =     0.4480

------------------------------------------------------------------------------
      status |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     bidders |  -.7553394   .3387904    -2.23   0.026    -1.419356   -.0913225
  difference |   .1122049   .0513911     2.18   0.029     .0114802    .2129296
       _cons |   1.421198   1.286747     1.10   0.269    -1.100781    3.943176
------------------------------------------------------------------------------
```

We see that both predictors are useful in predicting bid status. The estimates in Stata are obtained using maximum likelihood estimation and the tests for coefficients are Wald tests. The interpretation of them is no different than usual. The *predict* command still works with logistic regression, and in this case provides the estimated probability of the bid being fixed for a given set of predictors.

```
. predict yhat
(option pr assumed; Pr(status))
```

You can look in the browser to see the values of "yhat." The other commands we have used for postestimation analyses will also work. Note that the command *logistic* does the same analysis, but provides odds ratios in place of beta coefficient for ease of interpretation.

```
. logistic status bidders difference

Logistic regression                                     Number of obs   =         31
                                                        LR chi2(2)      =      18.54
                                                        Prob > chi2     =     0.0001
Log likelihood = -11.421512                             Pseudo R2       =     0.4480

------------------------------------------------------------------------------
      status | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     bidders |   .4698511    .159181    -2.23   0.026     .2418696    .9127233
  difference |   1.118742   .0574934     2.18   0.029     1.011546    1.237298
------------------------------------------------------------------------------
```