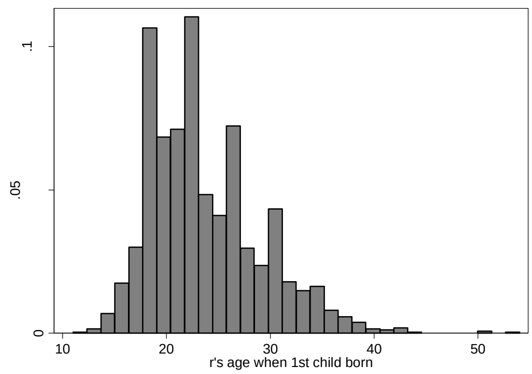
Sociology 7704: Regression Models for Categorical Data
Instructor: Natasha Sarkisian

Preliminary Data Screening

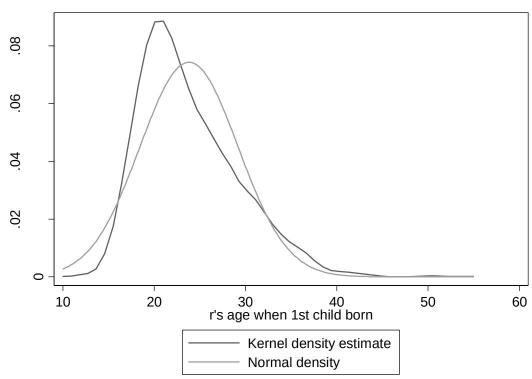
A. Examining Univariate Normality

Normality of each of the variables used in your model is not required, but it can often help us prevent further problems (especially heteroscedasticity and multivariate normality violations). Normality of the dependent variable is especially influential. We can examine the distribution graphically:

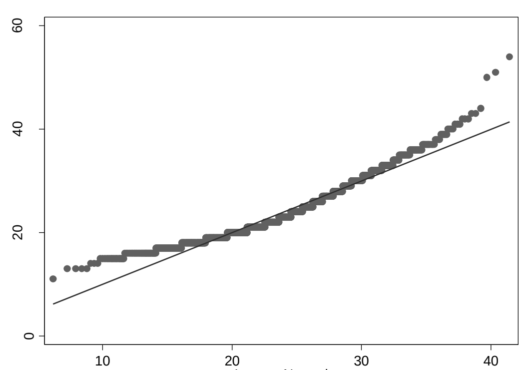
```
. histogram agekdbrn, normal
(bin=34, start=18, width=2.0882353)
```




```
. kdensity age, normal
```

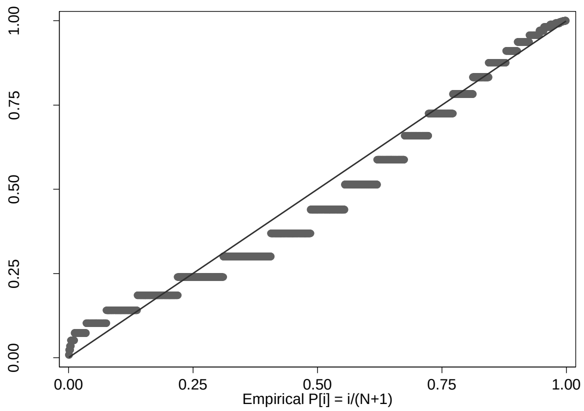



```
. qnorm agekdbrn
```



This is a quantile-normal (Q-Q) plot. It plots the quantiles of a variable against the quantiles of a normal distribution. In a perfectly normal distribution, all observations would be on the line, so the closest they are to being on the line, the closer the distribution to being normal. Any large deviations from the straight line indicate problems with normality. Note: this plot has nothing to do with linearity!

```
. pnorm agekdbrn
```



This is a standardized normal probability (P-P) plot, it is more sensitive to non-normality in the middle range of data, while qnorm is sensitive to non-normality near the tails.

We can also formally evaluate the distribution of a variable -- i.e., test the hypothesis of normality (with separate tests for skewness and kurtosis) using sktest:

```
. sktest age
                   Skewness/Kurtosis tests for Normality
                                                   ------- joint ------
    Variable |  Pr(Skewness)   Pr(Kurtosis)  adj chi2(2)    Prob>chi2
-------------+-------------------------------------------------------
         age |      0.000         0.000              .        0.0000
```

Here, the dot instead of chi-square value indicates that it's a very large number. This test is very sensitive to sample size, however – with large sample sizes, even small deviations from normality can be identified as statistically significant. But in this case, the graphs also confirmed this conclusion. Next, we'll consider transformations to bring this variable closer to normal. To search for transformations, we can use ladder command:

```
. ladder agekdbrn
Transformation         formula               chi2(2)       P(chi2)
------------------------------------------------------------------
cubic                  agekdbrn^3                  .         0.000
square                 agekdbrn^2                  .         0.000
identity               agekdbrn                    .         0.000
square-root            sqrt(agekdbrn)              .         0.000
log                    log(agekdbrn)           32.49         0.000
reciprocal root        1/sqrt(agekdbrn)         8.57         0.014
reciprocal             1/agekdbrn              14.84         0.001
reciprocal square      1/(agekdbrn^2)              .         0.000
reciprocal cubic       1/(agekdbrn^3)              .         0.000
```

Ladder allows you to search for normalizing transformation – the larger the P value, the closer to normal. Typically, square roots, log, and inverse (1/x) transformations normalize right (positive) skew. Inverse (reciprocal) transforms are “stronger” than logarithmic, which are “stronger” than square roots. For negative skews, we can use square or cubic transformation.

In this output, again, dots instead of chi2 indicate very large numbers. If there is a dot instead of P as well, it means that this specific transformation is not possible because of zeros or negative values. If zeros or negative values preclude a transformation that you think might help, the typical practice is to first add a constant that would get rid of such values (e.g., if you only have zeros but no negative values, you can add 1), and then perform a transformation. In this case, it appears that 1/square root brings the distribution closer to normal.

Note that just as sktest, in large samples the ladder command tests are rather sensitive to non-normalities – often it can be useful to take a random subsample and run ladder command on them to identify the best transformation. (But make sure the sample is not too small; keep it around 150-200 observations.)

```
. ladder age

Transformation         formula               chi2(2)       P(chi2)
------------------------------------------------------------------
cubic                  age^3                       .         0.000
square                 age^2                       .         0.000
identity               age                         .         0.000
square-root            sqrt(age)                   .         0.000
log                    log(age)                    .
0.000
reciprocal root        1/sqrt(age)                 .         0.000
reciprocal             1/age                       .         0.000
reciprocal square      1/(age^2)                   .         0.000
reciprocal cubic       1/(age^3)                   .         0.000
```

It’s not normal and none of the transformations seem to help. If your sample size is large, everything will be significantly different from normal, so you should either rely on graphical examination (gladder) or randomly select a subsample of your dataset and do this type of analysis for that subsample. We can use sample command to take a 5% random sample from the data. We first “preserve” the dataset so that we can bring the rest of observations back after we are done with ladder, and then sample:

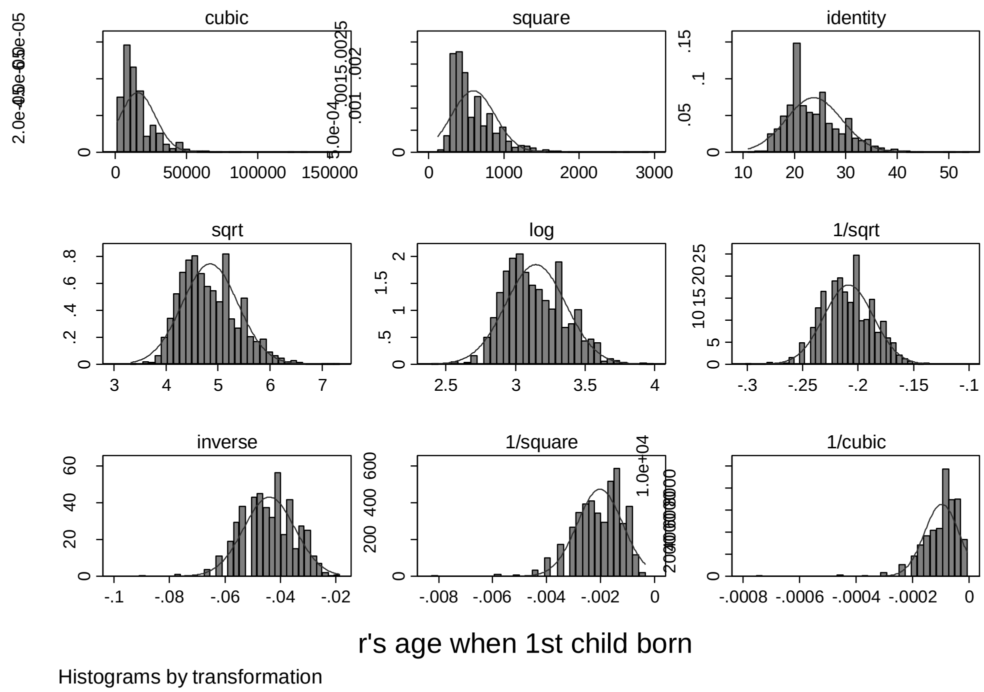
```
. preserve

. sample 5
(2627 observations deleted)

. ladder age
Transformation         formula               chi2(2)       P(chi2)
------------------------------------------------------------------
cubic                  age^3                   40.17         0.000
square                 age^2                   25.53         0.000
identity               age                     10.53         0.005
square-root            sqrt(age)                6.81         0.033
log                    log(age)                 5.99         0.050
reciprocal root        1/sqrt(age)              4.78         0.091
reciprocal             1/age                    8.23         0.016
reciprocal square      1/(age^2)               32.80         0.000
reciprocal cubic       1/(age^3)               63.69         0.000
```

Note that now it's much more clear which transformations bring this variable the closest to normal.

```
. restore
```

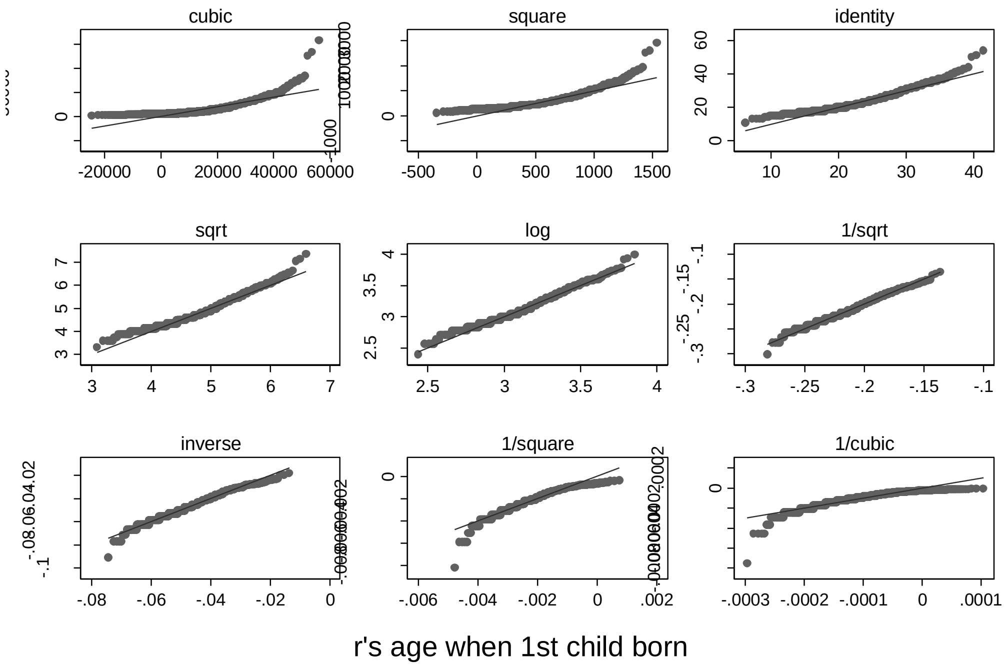
Restore command restores our original dataset (as it was when we ran preserve).
Let's examine transformations for agekdbrn graphically as well:

```
. gladder agekdbrn
```



Same using quantile-normal plots:

```
. qladder agekdbrn
```



Quantile-Normal plots by transformation

Let's attempt to use this transformation in our regression model:

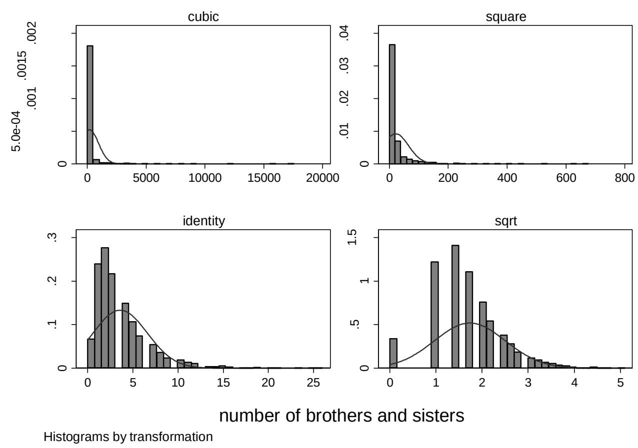
```
. gen agekdbrnrr=1/(sqrt(agekdbrn))
(810 missing values generated)
. reg agekdbrnrr educ born sex mapres80 age
      Source |       SS       df       MS              Number of obs =    1089
-------------+------------------------------           F(  5,  1083) =   54.00
       Model |  .107910937     5  .021582187           Prob > F      =  0.0000
    Residual |  .432834805  1083  .000399663           R-squared     =  0.1996
-------------+------------------------------           Adj R-squared =  0.1959
       Total |  .540745743  1088  .000497009           Root MSE      =  .01999
------------------------------------------------------------------------------
  agekdbrnrr |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |  -.0026108   .0002316   -11.27   0.000    -.0030652   -.0021564
        born |  -.0075379   .0023762    -3.17   0.002    -.0122004   -.0028755
         sex |   .0098921   .0012561     7.88   0.000     .0074274    .0123568
    mapres80 |  -.0001494    .000049    -3.05   0.002    -.0002455   -.0000533
         age |  -.0002532   .0000409    -6.19   0.000    -.0003336   -.0001729
       _cons |   .2535923   .0051683    49.07   0.000     .2434514    .2637332
------------------------------------------------------------------------------
```

Overall, transformations should be used sparsely - always consider ease of model interpretation as well. Here, our transformation made interpretation more complicated. It is also important to check that we did not introduce any nonlinearities by this transformation – we'll deal with that issue soon.

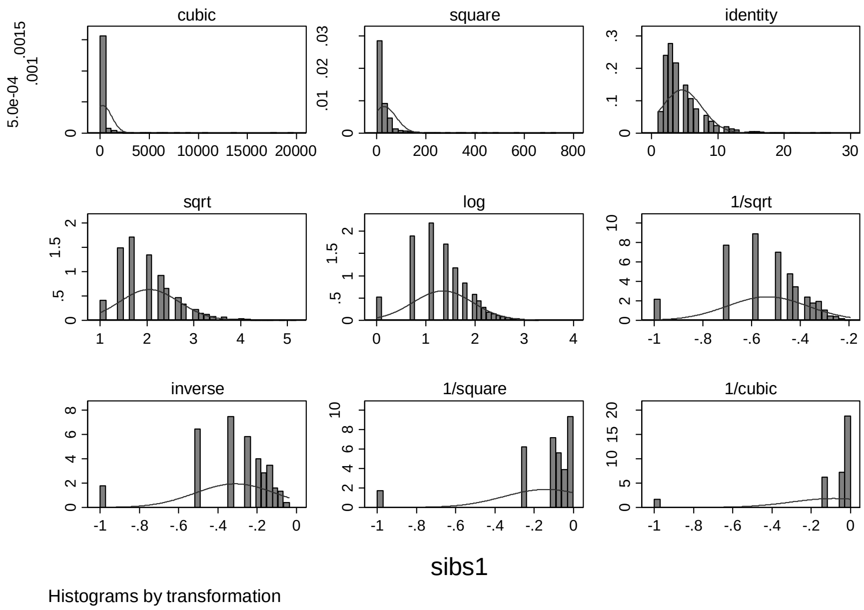
If a variable contains zero or negative values, you need to add a constant to it before looking for transformations (such that all values of the variable become larger than zero). For example:

```
. sum sibs
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        sibs |      2756    3.599419    2.997262          0         26

. ladder sibs

Transformation         formula                   chi2(2)       P(chi2)
------------------------------------------------------------------
cubic                  sibs^3                          .             .
square                 sibs^2                          .             .
identity               sibs                            .         0.000
square root            sqrt(sibs)                  64.41         0.000
log                    log(sibs)                       .             .
1/(square root)        1/sqrt(sibs)                    .             .
inverse                1/sibs                          .             .
1/square               1/(sibs^2)                      .             .
1/cubic                1/(sibs^3)                      .             .

. gladder sibs
```




```
. gen sibs1=sibs+1
(9 missing values generated)

. ladder sibs1

Transformation         formula                   chi2(2)       P(chi2)
------------------------------------------------------------------
cubic                  sibs1^3                         .             .
square                 sibs1^2                         .             .
identity               sibs1                           .         0.000
square root            sqrt(sibs1)                     .         0.000
log                    log(sibs1)                   0.48         0.787
1/(square root)        1/sqrt(sibs1)                   .         0.000
inverse                1/sibs1                         .         0.000
1/square               1/(sibs1^2)                     .             .
1/cubic                1/(sibs1^3)                     .             .

. gladder sibs1
```



If as variable is negatively skewed, you might have an easier time finding a transformation for it after reversing it. For this example, we will generate a scale of happiness that's the reverse of unhappiness scale and examine both distributions:

```
. tab1 happy7 satjob7 satfam7
```

-> tabulation of happy7

happy or unhappy on the whole	Freq.	Percent	Cum.
completely happy	141	12.16	12.16
very happy	510	43.97	56.12
fairly happy	391	33.71	89.83
neither happy nor unhappy	69	5.95	95.78
fairly unhappy	32	2.76	98.53
very unhappy	16	1.38	99.91
completely unhappy	1	0.09	100.00
Total	1,160	100.00	

-> tabulation of satjob7

job satisfaction in general	Freq.	Percent	Cum.
completely satisfied	127	15.49	15.49
very satisfied	289	35.24	50.73
fairly satisfied	264	32.20	82.93
neither satisfied nor dissatisfied	53	6.46	89.39
fairly dissatisfied	47	5.73	95.12
very dissatisfied	29	3.54	98.66
completely dissatisfied	11	1.34	100.00
Total	820	100.00	

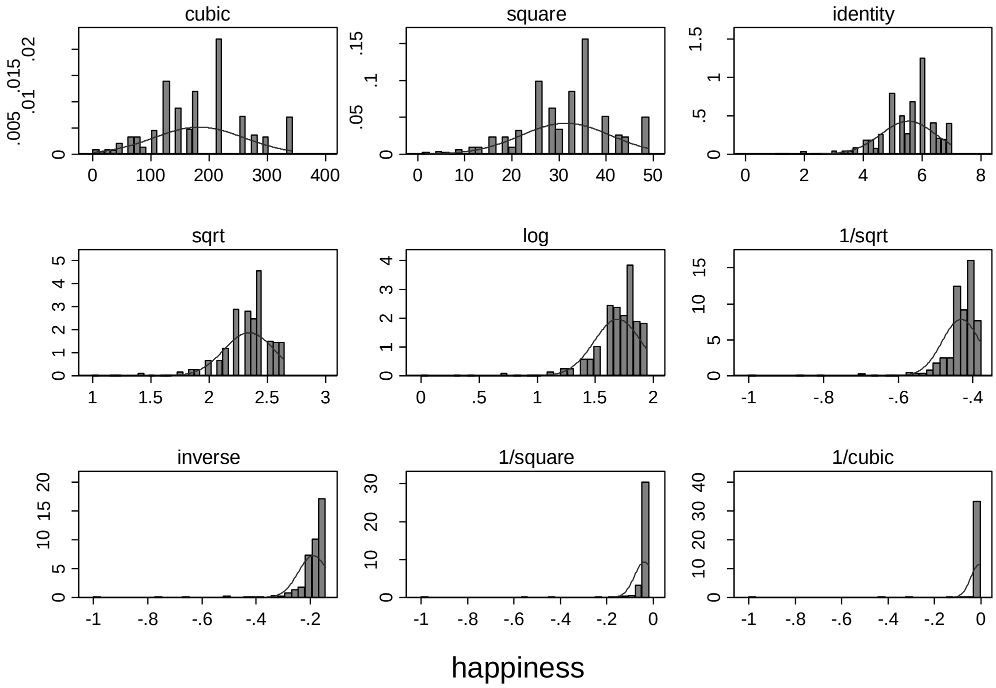
-> tabulation of satfam7

family satisfaction in general	Freq.	Percent	Cum.

```
----------------------------------+-----------------------------------
             completely satisfied |        265       23.08       23.08
                   very satisfied |        467       40.68       63.76
                 fairly satisfied |        286       24.91       88.68
neither satisfied nor dissatisfied |         70        6.10       94.77
              fairly dissatisfied |         31        2.70       97.47
                very dissatisfied |         20        1.74       99.22
          completely dissatisfied |          9        0.78      100.00
----------------------------------+-----------------------------------
                            Total |      1,148      100.00

. alpha happy7 satjob7 satfam7

Test scale = mean(unstandardized items)

Average interitem covariance:      .525359
Number of items in the scale:            3
Scale reliability coefficient:      0.6732

. egen unhappiness=rowmean(happy7 satjob7 satfam7)
(1600 missing values generated)

. sum unhappiness

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
 unhappiness |      1165    2.469814    .9298462          1          7
```

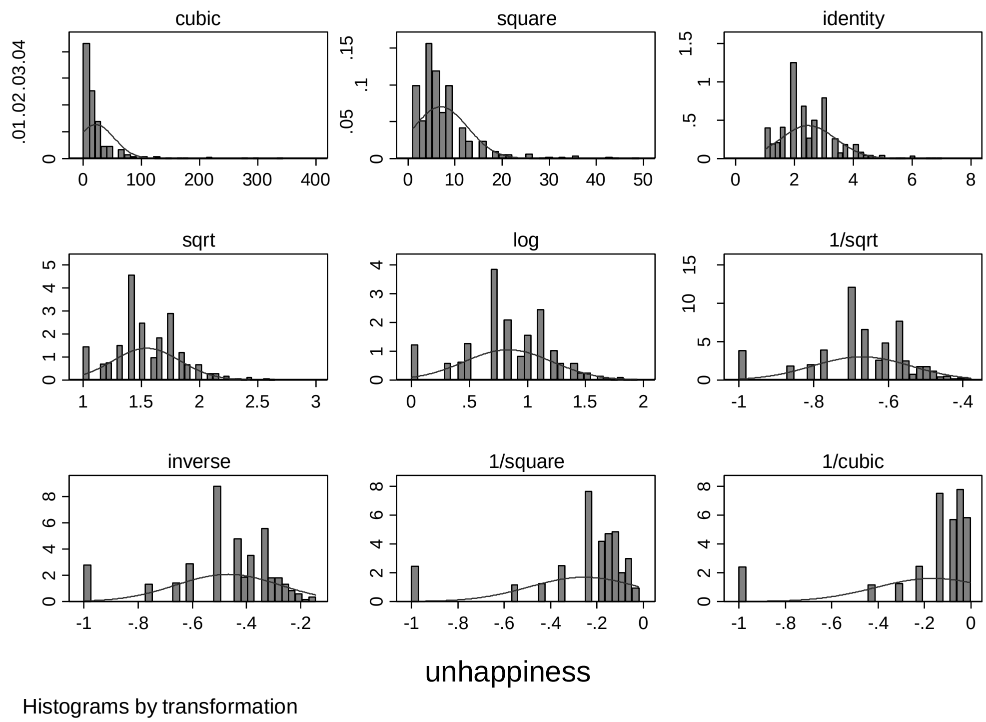
To reverse the scale, we add its maximum and its minimum and subtract the original scale from that:

```
. gen happiness=r(max)+r(min)-unhappiness
(1600 missing values generated)

. sum happiness

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
   happiness |      1165    5.530186    .9298462          1          7

. ladder happiness

Transformation         formula               chi2(2)       P(chi2)
------------------------------------------------------------------
cubic                  happin~s^3             11.17         0.004
square                 happin~s^2             15.55         0.000
identity               happin~s                   .         0.000
square root            sqrt(happin~s)             .         0.000
log                    log(happin~s)              .         0.000
1/(square root)        1/sqrt(happin~s)           .         0.000
inverse                1/happin~s                 .             .
1/square               1/(happin~s^2)             .             .
1/cubic                1/(happin~s^3)             .             .

. gladder happiness
```



Histograms by transformation

```
. ladder unhappiness

Transformation         formula                chi2(2)       P(chi2)
------------------------------------------------------------------
cubic                  unhapp~s^3                   .         0.000
square                 unhapp~s^2                   .         0.000
identity               unhapp~s                     .         0.000
square root            sqrt(unhapp~s)           27.32         0.000
log                    log(unhapp~s)            13.42         0.001
1/(square root)        1/sqrt(unhapp~s)             .         0.000
inverse                1/unhapp~s                   .         0.000
1/square               1/(unhapp~s^2)               .         0.000
1/cubic                1/(unhapp~s^3)               .         0.000

. gladder unhappiness
```



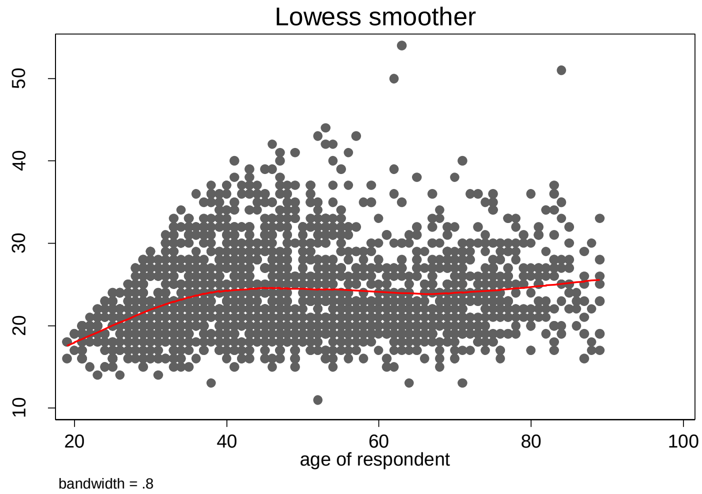
We might want to use log, but if we want the interpretation to be about happiness, we will reverse it again after transforming:

```
. gen unhappylog=log(unhappiness)
(1600 missing values generated)

. sum unhappylog

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
  unhappylog |      1165    .8345584    .3790659          0    1.94591

. gen unhappylogr=r(max)+r(min)-unhappylog
(1600 missing values generated)

. sum unhappylogr

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
 unhappylogr |      1165    1.111352    .3790659          0    1.94591
```

B. Examining bivariate linearity

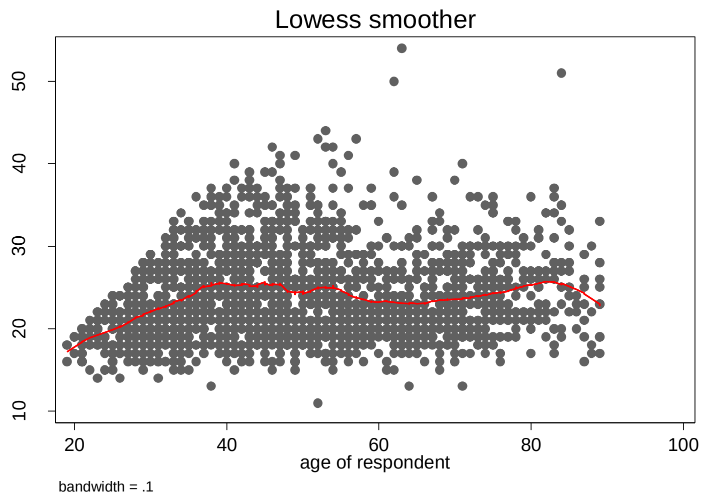
Before you run a regression, it’s a good idea to examine your variables one at a time as indicated before, but we should also examine the relationship of each independent variable to the dependent to assess its linearity. A good tool for such an examination is lowess – i.e., a scatterplot with a locally weighted regression line going through it (here, it is based on means, but we can also do it using medians):

```
. lowess agekdbrn age
```



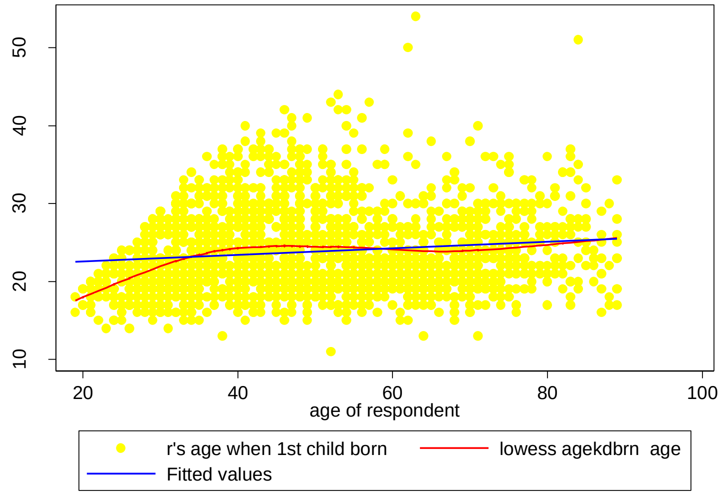
We can change bandwidth to make the curve less smooth (decrease the number) or smoother (increase the number):

```
. lowess agekdbrn age, bwidth(.1)
```



We can also add a regression line to see the difference better:

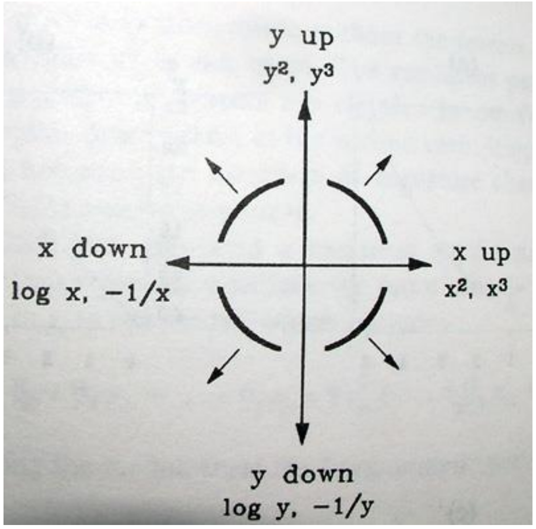
```
. scatter agekdbrn age, mcolor(yellow) || lowess agekdbrn age, lcolor(red) || lfit
agekdbrn age, lcolor(blue)
```



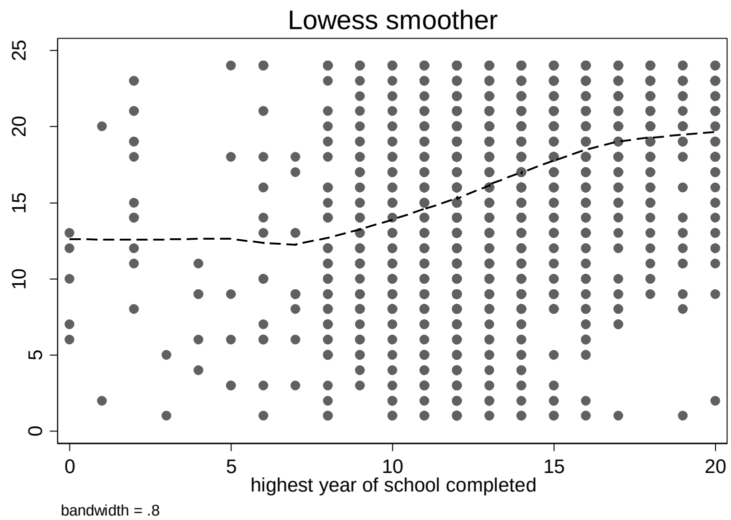
Based on lowess plots, we conclude that the relationship between age and agekdbrn is not linear and we need to address that.

Remedies for nonlinearity problems:
When we find a nonlinear relationship, we usually try to find a transformation to linearize it, although sometimes we may choose to break up the corresponding independent variable into a series of dummies instead.

1. <u>Monotone nonlinear relationship.</u> Power transformations can be used to linearize relationships if strong monotone nonlinearities are found. The following chart gives suggestions for transformations when the curve looks a certain way:

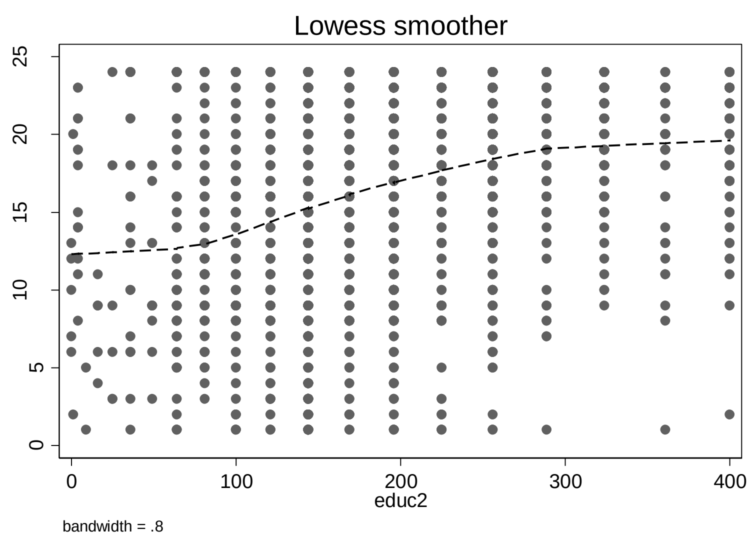



```
. lowess income98 educ
```

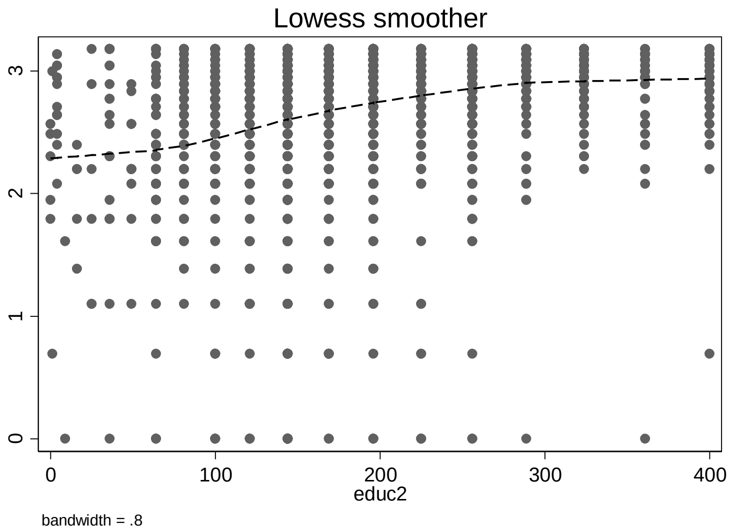


Either a square of X (educ) or a log of Y (income) should fix this.

```
. gen educ2=educ^2
(12 missing values generated)

. lowess income98 educ2
```




```
. gen income98lg=log(income98)
(121 missing values generated)

. lowess income98lg educ2
```



<u>2. Nonmonotone relationship.</u> For non-monotone relationships (e.g. parabola or cubic), use polynomial functions of the variable, e.g. age and age squared, etc. The pictures above for age and agekdbrn relationship would suggest that we might want to add a cubic term for age as well as a squared term. It is important, however, to attempt to maintain simplicity and interpretability of the results when doing transformations. So let's try squared term. We want to enter both age and age squared into our regression model. But using age and age squared in the model at the same time will create multicollinearity because the two variables have a strong relationship—to avoid that, we have to mean-center age prior to generating a square and a cube. That is, whenever we plan to use more than a single term for the same variable in our regression model, always mean-center (i.e., if you just plan to use age squared without age, like we did for educ in the example above, then you don't need to mean center, but if we wanted to use both educ and educ2, we'd have to mean-center educ and only then generate educ2).

For example, without mean-centering:

```
. gen age2=age^2
(14 missing values generated)

. reg agekdbrn educ born sex mapres80 age age2

      Source |       SS       df       MS              Number of obs =    1089
-------------+------------------------------           F(  6,  1082) =   44.22
       Model |  6138.53315     6  1023.08886           Prob > F      =  0.0000
    Residual |  25034.1298  1082  23.1369037           R-squared     =  0.1969
-------------+------------------------------           Adj R-squared =  0.1925
       Total |   31172.663  1088  28.6513447           Root MSE      =  4.8101
------------------------------------------------------------------------------
    agekdbrn |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .5678949   .0569661     9.97   0.000     .4561184    .6796713
        born |   1.567736   .5723843     2.74   0.006     .4446266    2.690844
         sex |  -2.140989   .3028244    -7.07   0.000    -2.735179   -1.546799
    mapres80 |   .0332034   .0117896     2.82   0.005     .0100704    .0563364
         age |   .2808181    .055909     5.02   0.000     .1711158    .3905203
        age2 |  -.0022448   .0005551    -4.04   0.000     -.003334   -.0011556
       _cons |    8.92424   1.643755     5.43   0.000     5.698932    12.14955
------------------------------------------------------------------------------
```

```
. reg  agekdbrn educ born sex mapres80 age age2, beta
      Source |       SS       df       MS              Number of obs =    1089
-------------+------------------------------           F(  6,  1082) =   44.22
       Model |  6138.53315     6  1023.08886           Prob > F      =  0.0000
    Residual |  25034.1298  1082  23.1369037           R-squared     =  0.1969
-------------+------------------------------           Adj R-squared =  0.1925
       Total |   31172.663  1088  28.6513447           Root MSE      =  4.8101
------------------------------------------------------------------------------
    agekdbrn |      Coef.   Std. Err.      t    P>|t|                     Beta
-------------+----------------------------------------------------------------
        educ |   .5678949   .0569661     9.97   0.000                 .2884756
        born |   1.567736   .5723843     2.74   0.006                 .0751117
         sex |  -2.140989   .3028244    -7.07   0.000                -.1937892
    mapres80 |   .0332034   .0117896     2.82   0.005                  .080348
         age |   .2808181    .055909     5.02   0.000                  .790523
        age2 |  -.0022448   .0005551    -4.04   0.000                 -.637722
       _cons |    8.92424   1.643755     5.43   0.000                        .
------------------------------------------------------------------------------
```

Note that age and age2 have high betas with opposite signs -- that's one indication of multicollinearity. Often when high degree of multicollinearity is present, we would also observe high standard errors. In fact, when reading published research using OLS, pay attention to standard errors -- if they are high relative the to size of the coefficient itself, it's a reason for a concern about possible multicollinearity. Let's check our suspicion using VIFs (Variance Inflation Factors):

```
. vif
    Variable |       VIF       1/VIF
-------------+----------------------
        age2 |     33.51    0.029845
         age |     33.37    0.029963
        educ |      1.13    0.886374
    mapres80 |      1.10    0.911906
        born |      1.01    0.986930
         sex |      1.01    0.987914
-------------+----------------------
    Mean VIF |     11.86
```

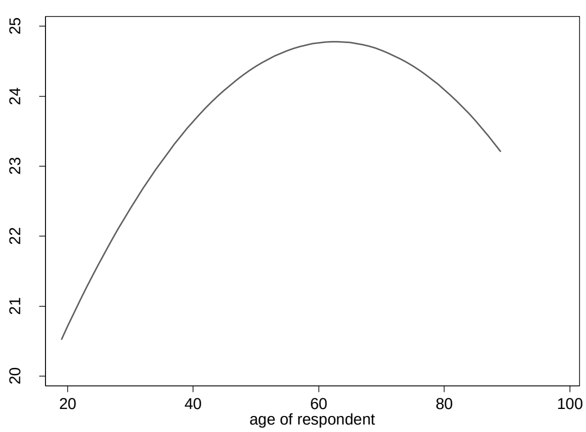
Indeed, high degree of multicollinearity. But luckily, we can avoid it. When including variables that are generated using other variables already in the model (as in this case, or when we want to enter a product of two variables to model an interaction term), we should first mean-center the variable (only if it is continuous; don't mean-center dichotomous variables!). That's how we'd do it in this case:

```
. sum age
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         age |      2751    46.28281    17.37049         18         89
. gen agemean=age-r(mean)
(14 missing values generated)
. gen agemean2=agemean^2
(14 missing values generated)

. reg  agekdbrn educ born sex mapres80 agemean agemean2, beta
      Source |       SS       df       MS              Number of obs =    1089
-------------+------------------------------           F(  6,  1082) =   44.22
       Model |  6138.53316     6  1023.08886           Prob > F      =  0.0000
    Residual |  25034.1298  1082  23.1369037           R-squared     =  0.1969
-------------+------------------------------           Adj R-squared =  0.1925
       Total |   31172.663  1088  28.6513447           Root MSE      =  4.8101
------------------------------------------------------------------------------
    agekdbrn |      Coef.   Std. Err.      t    P>|t|                     Beta
```

```
-------------+----------------------------------------------------------------
        educ |   .5678949   .0569661     9.97   0.000                 .2884756
        born |   1.567736   .5723843     2.74   0.006                 .0751117
         sex |  -2.140989   .3028244    -7.07   0.000                -.1937892
    mapres80 |   .0332034   .0117896     2.82   0.005                  .080348
     agemean |   .0730284   .0105054     6.95   0.000                 .2055801
    agemean2 |  -.0022448   .0005551    -4.04   0.000                -.1209343
       _cons |   17.11274   1.126117    15.20   0.000                        .
------------------------------------------------------------------------------
. vif
    Variable |       VIF       1/VIF
-------------+----------------------
    agemean2 |      1.20    0.829918
     agemean |      1.18    0.848643
        educ |      1.13    0.886374
    mapres80 |      1.10    0.911906
        born |      1.01    0.986930
         sex |      1.01    0.987914
-------------+----------------------
    Mean VIF |      1.11
```

We can see that the multicollinearity problem has been solved. We also note that the squared term is significant. To better understand what this means substantively, we'll generate a graph:

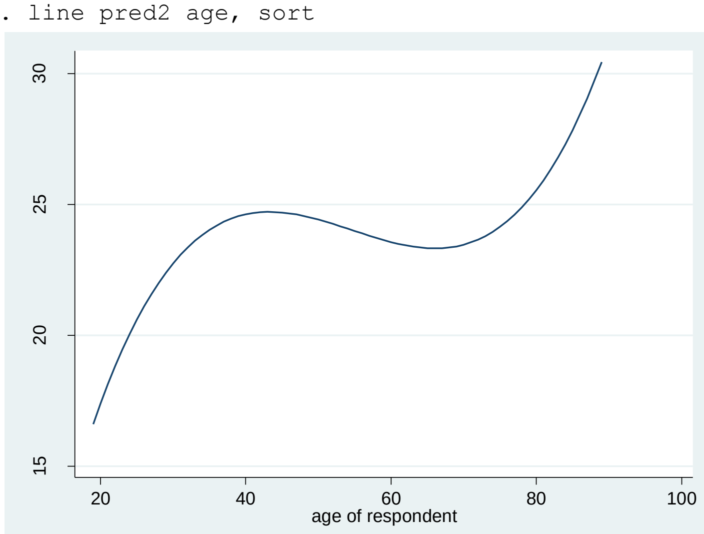
```
. adjust educ born sex mapres80 if e(sample), gen(pred1)
-------------------------------------------------------------------------------
     Dependent variable: agekdbrn     Command: regress
       Created variable: pred1
   Variables left as is: age, age2
 Covariates set to mean: educ = 13.316804, born = 1.0707071, sex = 1.6244261, mapres80
= 39.440773
-------------------------------------------------------------------------------
      All |         xb
----------+-----------
          |    23.6648
----------------------
     Key:  xb  =  Linear Prediction

. line pred1 age, sort
```



This doesn't quite replicate what we saw on lowess plot, so the relationship of age and agekdbrn is likely still misspecified. Let's try cube:

```
. gen agemean3=agemean^3
(14 missing values generated)
```

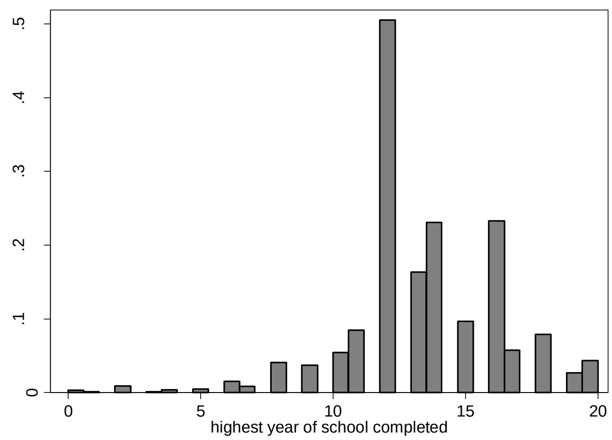
```
. reg  agekdbrn educ born sex mapres80 agemean agemean2 agemean3
      Source |       SS       df       MS              Number of obs =    1089
-------------+------------------------------           F(  7,  1081) =   49.39
       Model |  7554.31674     7  1079.18811           Prob > F      =  0.0000
    Residual |  23618.3463  1081  21.8486089           R-squared     =  0.2423
-------------+------------------------------           Adj R-squared =  0.2374
       Total |   31172.663  1088  28.6513447           Root MSE      =  4.6742
------------------------------------------------------------------------------
    agekdbrn |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |    .581195    .055382    10.49   0.000     .4725265    .6898634
        born |   1.292907   .5572673     2.32   0.021     .1994591    2.386355
         sex |  -2.117214   .2942876    -7.19   0.000    -2.694654   -1.539774
    mapres80 |   .0349051   .0114586     3.05   0.002     .0124215    .0573887
     agemean |  -.0424837   .0176105    -2.41   0.016    -.0770384    -.007929
    agemean2 |  -.0059131   .0007061    -8.37   0.000    -.0072987   -.0045275
    agemean3 |   .0002359   .0000293     8.05   0.000     .0001784    .0002934
       _cons |   17.58535    1.09589    16.05   0.000     15.43504    19.73566
------------------------------------------------------------------------------

. adjust educ born sex mapres80 if e(sample), gen(pred2)
-------------------------------------------------------------------------------
     Dependent variable: agekdbrn     Command: regress
       Created variable: pred2
   Variables left as is: agemean, agemean2, agemean3
 Covariates set to mean: educ = 13.316804, born = 1.0707071, sex = 1.6244261, mapres80
= 39.440771
-------------------------------------------------------------------------------
      All |         xb
----------+-----------
          |    23.6648
----------------------
     Key:  xb  =  Linear Prediction

. line pred2 age, sort
```



C. Screening for Univariate and Bivariate Outliers

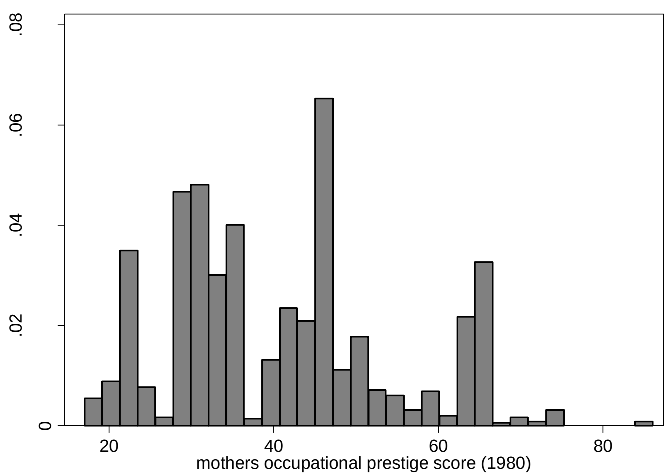
We usually start identifying potential outliers when conducting univariate and bivariate examination of the data. For example, when examining the distribution of educ, we would be concerned about those with very few years of education:

```
. histogram educ
```



When examining the distribution of mother's prestige, we'd be concerned about those with very high values:

```
. histogram mapres80
```



Such observations are likely high leverage points and we might want to deal with them early on, oftentimes by topcoding or bottomcoding:

```
. tab educ
    highest |
    year of |
     school |
  completed |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |          5        0.18        0.18
          1 |          2        0.07        0.25
          2 |         15        0.54        0.80
          3 |          2        0.07        0.87
```

```
          4 |          6        0.22        1.09
          5 |          8        0.29        1.38
          6 |         25        0.91        2.29
          7 |         14        0.51        2.80
          8 |         66        2.40        5.19
          9 |         60        2.18        7.37
         10 |         88        3.20       10.57
         11 |        137        4.98       15.55
         12 |        818       29.71       45.26
         13 |        265        9.63       54.89
         14 |        374       13.59       68.47
         15 |        157        5.70       74.17
         16 |        377       13.69       87.87
         17 |         93        3.38       91.25
         18 |        128        4.65       95.90
         19 |         43        1.56       97.46
         20 |         70        2.54      100.00
------------+-----------------------------------
      Total |      2,753      100.00

. gen educb=educ
(12 missing values generated)

. drop educb

. gen educb7=educ
(12 missing values generated)

. replace educb7=7 if educ<7
(63 real changes made)

. tab educb7

     educb7 |      Freq.     Percent        Cum.
------------+-----------------------------------
          7 |         77        2.80        2.80
          8 |         66        2.40        5.19
          9 |         60        2.18        7.37
         10 |         88        3.20       10.57
         11 |        137        4.98       15.55
         12 |        818       29.71       45.26
         13 |        265        9.63       54.89
         14 |        374       13.59       68.47
         15 |        157        5.70       74.17
         16 |        377       13.69       87.87
         17 |         93        3.38       91.25
         18 |        128        4.65       95.90
         19 |         43        1.56       97.46
         20 |         70        2.54      100.00
------------+-----------------------------------
      Total |      2,753      100.00

. sum mapres80

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
    mapres80 |      1619    40.96912    13.63189         17         86

. tab mapres80

    mothers |
occupationa |
 l prestige |
```

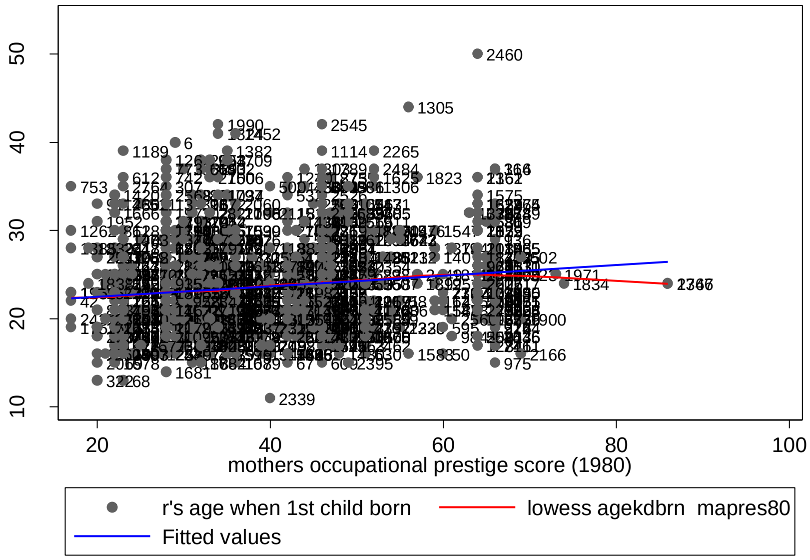
score (1980)	Freq.	Percent	Cum.
17	14	0.86	0.86
19	5	0.31	1.17
20	22	1.36	2.53
21	9	0.56	3.09
22	39	2.41	5.50
23	83	5.13	10.62
24	14	0.86	11.49
25	13	0.80	12.29
26	3	0.19	12.48
27	3	0.19	12.66
28	125	7.72	20.38
29	38	2.35	22.73
30	28	1.73	24.46
31	53	3.27	27.73
32	87	5.37	33.11
33	57	3.52	36.63
34	48	2.96	39.59
35	63	3.89	43.48
36	77	4.76	48.24
37	1	0.06	48.30
38	4	0.25	48.55
39	16	0.99	49.54
40	30	1.85	51.39
41	5	0.31	51.70
42	77	4.76	56.45
43	21	1.30	57.75
44	39	2.41	60.16
45	13	0.80	60.96
46	160	9.88	70.85
47	68	4.20	75.05
48	9	0.56	75.60
49	30	1.85	77.46
50	2	0.12	77.58
51	60	3.71	81.28
52	19	1.17	82.46
53	6	0.37	82.83
54	10	0.62	83.45
55	11	0.68	84.13
56	4	0.25	84.37
57	7	0.43	84.81
59	8	0.49	85.30
60	16	0.99	86.29
61	7	0.43	86.72
63	2	0.12	86.84
64	74	4.57	91.41
65	14	0.86	92.28
66	100	6.18	98.46
67	1	0.06	98.52
68	1	0.06	98.58
69	6	0.37	98.95
73	3	0.19	99.14
74	9	0.56	99.69
75	2	0.12	99.81
86	3	0.19	100.00
Total	1,619	100.00	

```
. gen  mapres80t66=mapres80
(1146 missing values generated)
```

```
. replace mapres80t66=66 if mapres80>66 & mapres80<.
(25 real changes made)
```

Bivariate examination can further help us identify potential leverage points and outliers. For example, we can label observations in the lowess plot to pinpoint problematic ones:

```
. scatter agekdbrn mapres80, mlabel(id) || lowess agekdbrn mapres80, lcolor(red) ||
lfit agekdbrn mapres80, lcolor(blue)
```



What we see standing out here is 2460 which has a high value on agekdbrn as well as two observations that have very high values of mother's prestige score; these are 2366 and 1747:

```
. list id mapres80 if mapres80>80 & mapres80~=. & agekdbrn~=.
      +-----------------+
      |   id   mapres80 |
      |-----------------|
1896. | 2366         86 |
2447. | 1747         86 |
      +-----------------+
```

In this case, we would notice all of these on univariate plots as well, but sometimes, we do detect problematic observations on such plots that go beyond what we see in univariate ones.