Transformation of Variables

SPRM Section 7.18

- What do we do when linear regression assumptions are violated? Here, we will consider the situations of non-linearity and non-normality (heteroscedasticity (unequal Y variance over X) will be dealt with later).
- **Transformations** are useful tools – we transform (rescale, generally) the variables in the model so that the linear regression model becomes (more) appropriate.
- Transformations may simplify analyses, as using linear regression is much simpler than using a non-linear estimation procedures and subsequent inference. However, we cannot 'fix' all problems - that is, a non-linear model may be needed.

Transformation of Variables

- There are often many ways of transforming the variables in a model, and often not a single "right one". You might try more than one, and choose that which provides the right balance of model fit and ease of interpretation.
- **Remember** – whenever you transform your variables, all your estimates and confidence intervals are expressed in that scale. To report your results in the more interpretable original scale, you need to convert (i.e., transform) BACK to the original scale.

Transformation of Variables

- **Recall that all of the following can be considered linear models:**

$$Y = \beta_0 + \beta_1 X + \epsilon$$

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$$

$$Y = \beta_0 + \beta_1 \log_e(X) + \epsilon$$

Whereas the following are not linear models:

$$Y = \beta_0 + \exp(\beta_1 X) + \epsilon$$

$$Y = \Pr(D = 1) = \frac{\exp(\beta_0 + \beta_1 X)}{1 + \exp(\beta_0 + \beta_1 X)} + \epsilon$$

Transformation of Variables

- **Many nonlinear model forms can be 'linearized'.** If the context-specific model is nonlinear but can be transformed this way, then it permits all the methods of linear regression to be applied.
- Note that transformations may be applied to Y, X, or both, depending on the circumstance and purpose
- Sometimes the response variable Y has natural properties that violate regression assumptions other than normality. We use specific transformations to address this.
- **See Table 6.1 in C&H** for several transformation options

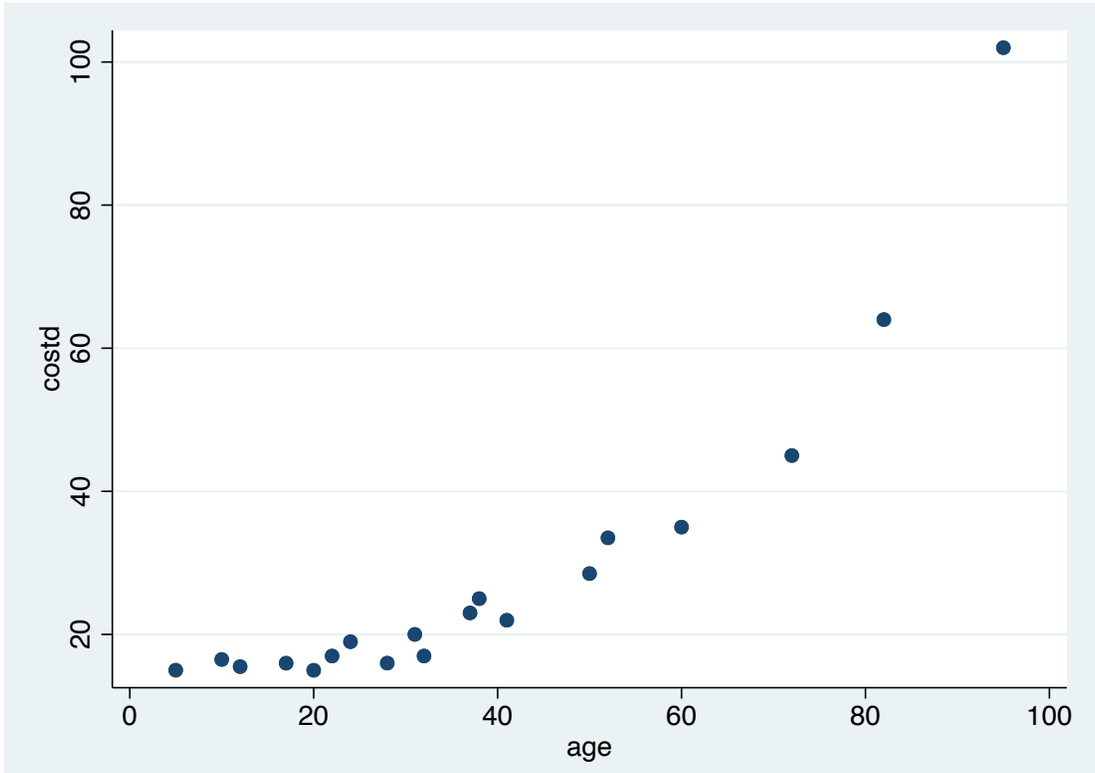
Common Transformations - an example

Example: Port Wine Price by Age

Port wine is generally more valuable with increasing age (vintage year). Below is a data listing of price and age (offset from a fixed date)

	age	costd
1.	82	64
2.	72	45
3.	52	33.5
4.	41	21.98
5.	38	25
6.	37	23
7.	32	16.98
8.	31	20
9.	28	15.99
10.	24	18.98
11.	22	16.98
12.	20	14.99
13.	17	15.98
14.	12	15.5
15.	95	102
16.	60	35

```
17.     10    16.5
18.      5      15
19.     50    28.5
```



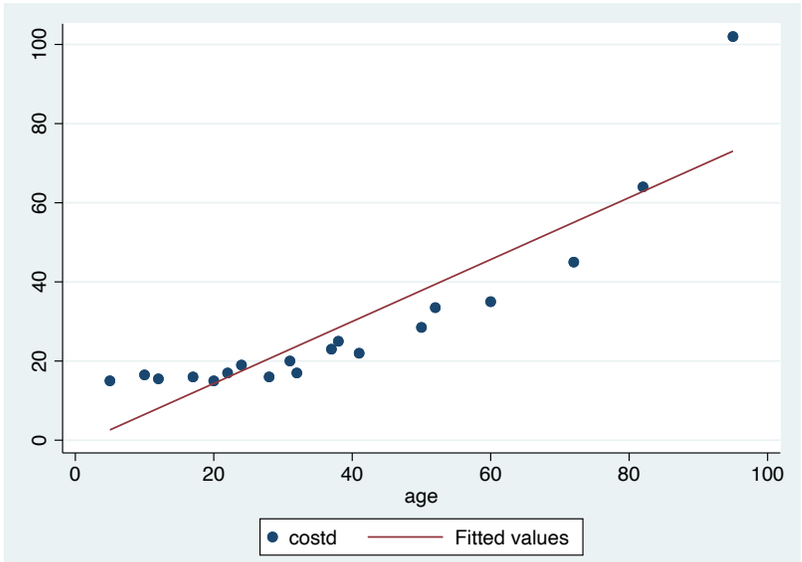
We want to predict price with vintage year or some function of it.

Common Transformations - an example

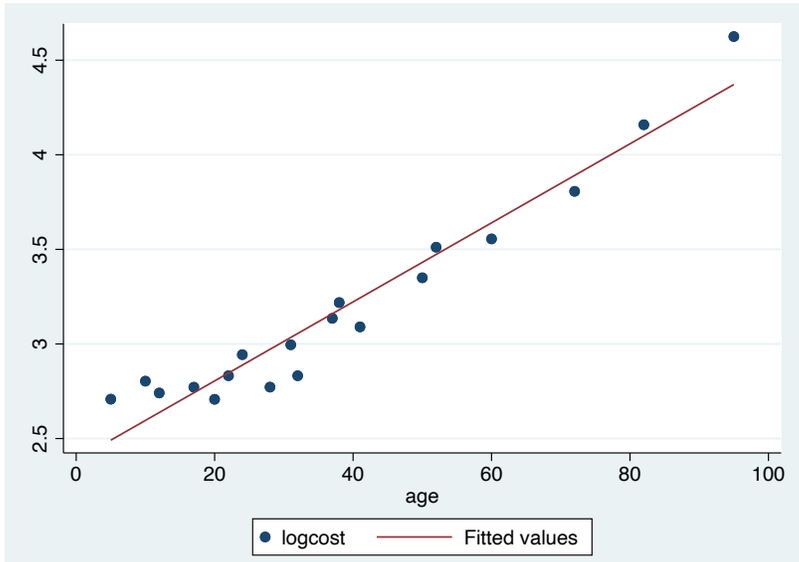
Do we have the right scale for Y (cost)? There is some curvature upwards, and we may try the logarithmic transform on Y. **Note:** default base in STATA is e, or natural log

```
. gen logcost = log(costd)
.
. twoway (scatter costd age) (lfit costd age)
. twoway (scatter logcost age) (lfit logcost age)
```

The transformed data looks improved with respect to linearity



(a) Original scale for price



(b) Natural log scale scale for price

Transformations example

The linear regression before transformation of Y

```
. reg costd age

      Source |       SS           df       MS      Number of obs   =        19
-------------+----------------------------------   F(1, 17)        =     68.55
       Model |  6815.55449         1  6815.55449   Prob > F        =    0.0000
    Residual |  1690.17649        17  99.4221465   R-squared       =    0.8013
-------------+----------------------------------   Adj R-squared   =    0.7896
       Total |  8505.73098        18   472.54061   Root MSE        =    9.9711
------------------------------------------------------------------------------
       costd |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .7827409   .0945386     8.28   0.000      .583282    .9821999
       _cons |  -1.313443   4.284151    -0.31   0.763    -10.35221    7.725325
------------------------------------------------------------------------------
```

Common Transformations - an example

- But how should we interpret this model? And what is the model now anyway? It is specified as:

$$\mathrm{E}(\log \mathrm{Price}) = \beta_0 + \beta_1 \mathrm{Age}$$

This means that the mean of log(Price) increases by a fixed amount for each increment in Age. What does this mean on the scale of Price (i.e. what does it mean in dollars)?

$$\exp(\mathrm{E}(\log \mathrm{Price})) = \exp(\beta_0)\exp(\beta_1 \mathrm{Age}).$$

More simply written:

$$\mathrm{E}(\mathrm{Price}) = ke^{(\beta_1 \mathrm{Age})}$$

where $k = \exp(\beta_0)$. So, there is a a nonlinear effect of age on price

Common Transformations - an example

If the age of Port increases by one year, from T to $(T+1)$,

$$\exp(\log \widehat{\text{Price}}_{old}) = \exp(\hat{\beta}_0)\exp(\hat{\beta}_1 T).$$

$$\exp(\log \widehat{\text{Price}}_{new}) = \exp(\hat{\beta}_0)\exp(\hat{\beta}_1 (T+1)).$$

Then

$$\widehat{\text{Price}}_{new} = \widehat{\text{Price}}_{old} \times \exp(\hat{\beta}_1).$$

The prices (in \$) would be expected to increase by a fixed multiple $\exp(\beta_1)$ per year. One could convert to % increase by $(\exp(\beta_1) - 1) \times 100\%$.

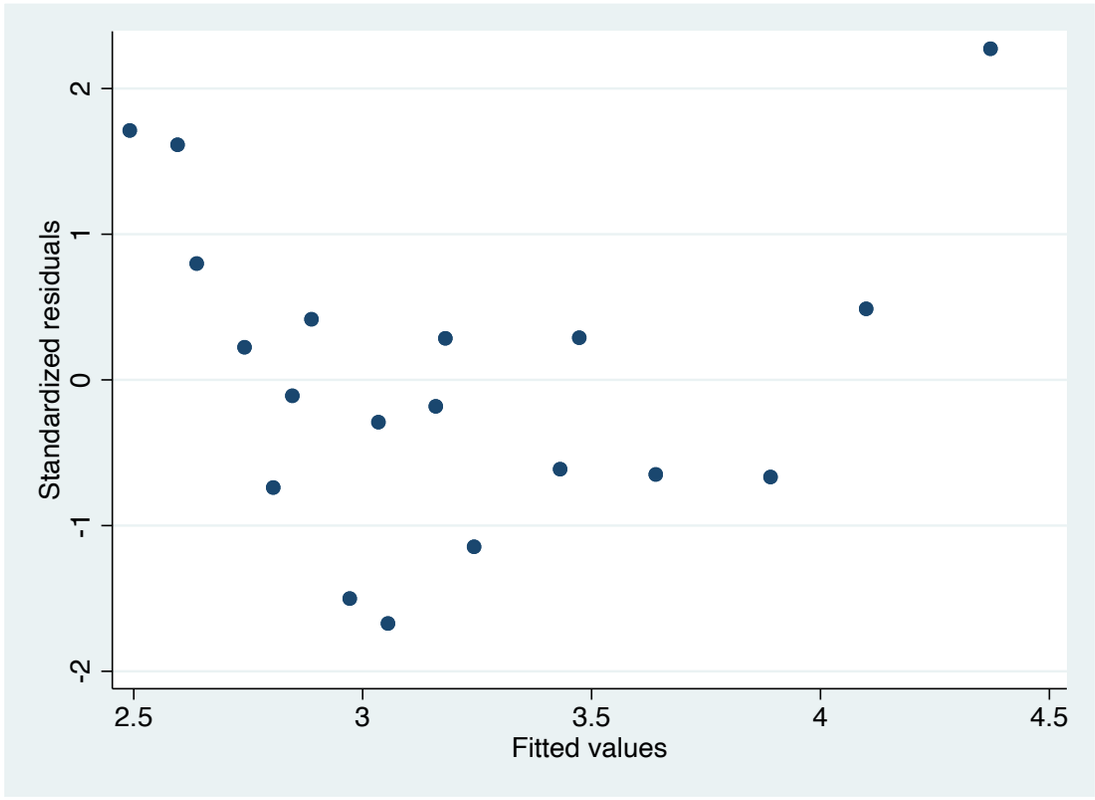
- Note that if one works on the log(price) scale, $(\hat{\beta}_0 + \hat{\beta}_1(T+1)) - (\hat{\beta}_0 + (\hat{\beta}_1(T)) = \beta_1$ or the increment in price on the log scale

Common Transformations - an example

The results for this regression:

```
. reg logcost age

      Source |       SS           df       MS      Number of obs   =        19
-------------+----------------------------------   F(1, 17)        =    256.81
       Model |  4.85418804         1  4.85418804   Prob > F        =    0.0000
    Residual |  .321337776        17  .018902222   R-squared       =    0.9379
-------------+----------------------------------   Adj R-squared   =    0.9343
       Total |  5.17552581        18  .287529212   Root MSE        =    .13749

------------------------------------------------------------------------------
     logcost |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .0208894   .0013035    16.03   0.000     .0181392    .0236396
       _cons |   2.386867   .0590717    40.41   0.000     2.262236    2.511497
------------------------------------------------------------------------------
```

The R^2 is higher. The standardized residuals seem not too bad, still maybe problems at extremes of age



The model also suggests that price increases by about 2% per year:

$$\exp(.0209) = 1.021$$

Note that for small x, $\exp(x)$ is approximately $1 + x$.

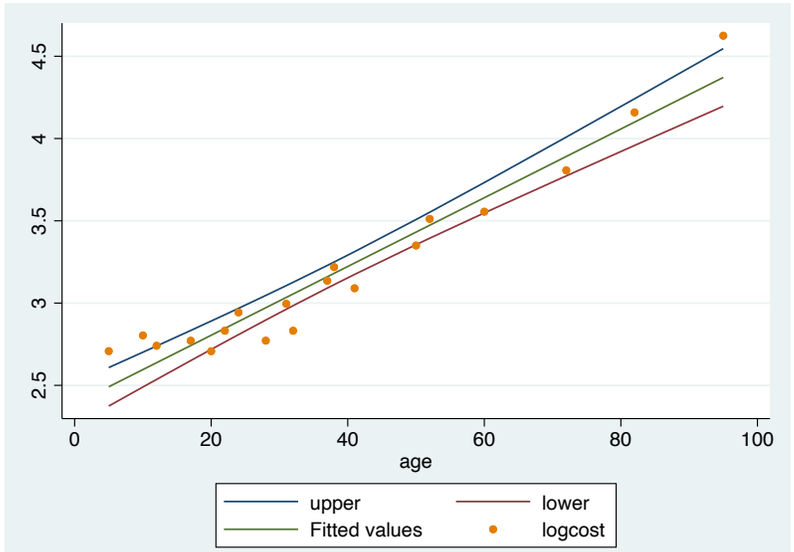
How do we do predictions with this model?

```
. predict sepred, stdp
. predict lphat
.
. sort age
. gen upper=lphat + invttail(12, 0.025) * sepred
. gen lower=lphat - invttail(12, 0.025) * sepred
. scatter upper lower lphat logprice age, c(l l l .) s(i i i o)
```

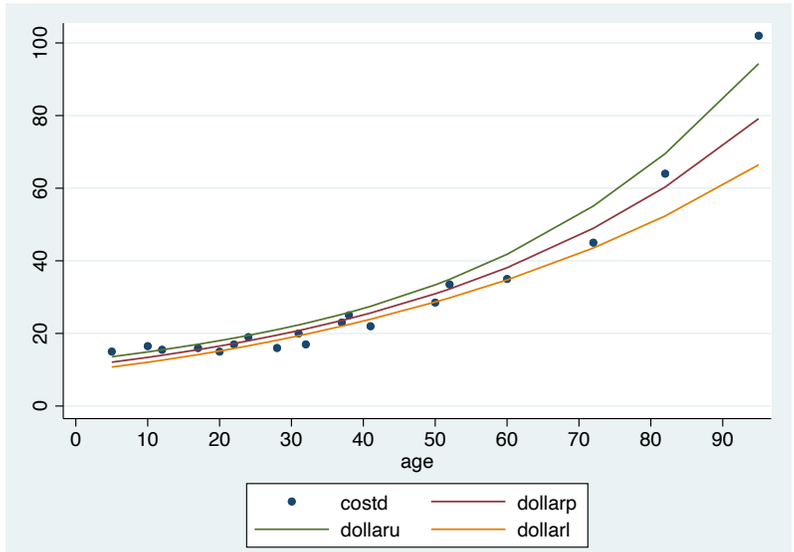
Converting back to the original model:

```
. gen dollarp=exp(lphat)
. gen dollaru=exp(upper)
. gen dollarl=exp(lower)
. scatter price dollarp dollaru dollarl age, s(o i i i) c(. l l l)
   xlabel(0(10)90) ylabel(0(20)100)
```

Note that once we convert back to the original scale, the confidence bands are no longer symmetric, and increasing in width.



(c) Log scale (in log-dollars)



(d) Original scale (in dollars)

Choosing a Transformation

How do we decide which transformation is most appropriate? We will specifically look at addressing violations of linearity, homoscedasticity, and normality of Y.

- Different types of violations call for different transformations, but it is often the case that the best transformations happen to address several violations at once.
- Non-normality of the response is usually the least serious violation, unless it is severe. Normality also happens to be the one problem usually remedied automatically after the other violations have been addressed.

Violations of Linearity - General Considerations

So how do we decide which transformation for correcting linearity is most appropriate?

- **Analytic approach:**
  If the context/theory suggests a non-linear model, we may try to linearize it analytically.

- **Empirical approach:**
  Often, we have no such structural model, but rather notice violations when working with the the data. We must decide which transformation, if any, to employ. For example, the "log" transformation used earlier is popular. It frequently works when a process/relationship is on a multiplicative scale (linear on a logarithmic scale)

Power Transformations

- In addition to log transformation, there are other ways to transform a variable:

- **Ladder Of Power:** Transforming Y by exponentiating it by some quantity
  - -1 - reciprocal
  - $-0.5, 0.5, .333,$ - root transforms (square root, third root, etc)
  - $1, 2, 3,$etc. - powers

- **Box-Cox Transformation:** Transforming Y into Y' via

$$Y' = (Y^\lambda - 1)/\lambda$$

  This is a general approach that includes several power transformations (and log transforms) as special cases. We will come back to this shortly.

Transformations

- Sometimes, one must consider the special form of Y and how it deviates from normality. Depending on Y, a specific transform may be suggested based on statistical theory considerations. This depends on knowing some 'tricks of the trade'

  Ex: Airline Injury Data, see C &HTable 6.6. Y is the number of injury incidents, X is the proportion of total flights across airlines.

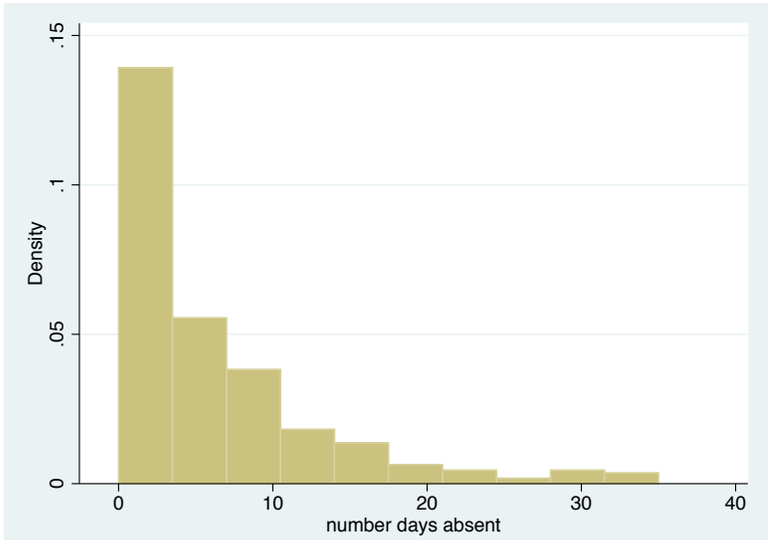
  - The response here is a count of a relatively rare event, and thus may be distributed as *Poisson* rather than normal. In the Poisson, the variance is proportional to the mean (not independent of the mean as in normal dist'n). So, <u>if the mean of Y changes over X</u> (i.e., regression model is meaningful), we have an implicit regression assumption violation
  - It can be shown that $\sqrt{Y}$ will have more stable variance over the range of Y.

Transformations

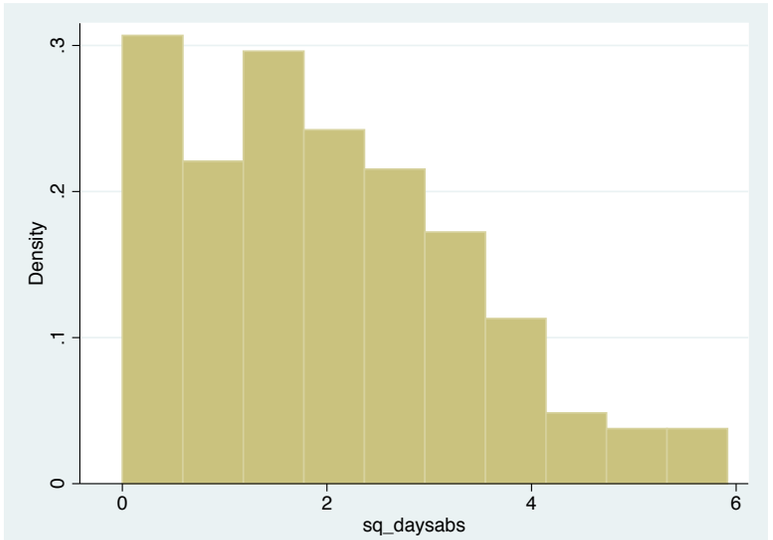
Another example: School days absent at two junior high schools predicted by factors including standardized math score. (http://www.ats.ucla.edu/stat/stata/dae/nbreg.htm). The distribution of days is highly skewed (not normal). We will transform it.

```
. gen sq_daysabs = sqrt(daysabs)
.*  look at data
. hist daysabs, bin(10)
(bin=10, start=0, width=3.5)
. hist sq_daysabs, bin(10)
(bin=10, start=0, width=.591608)
’
.* make scatter plot with fitted regression line
. twoway (scatter daysabs math) (lfit daysabs math)
. twoway (scatter sq_daysabs math) (lfit sq_daysabs math)
```

Histograms of days absent - transform is a bit better - but not gaussian (normal)

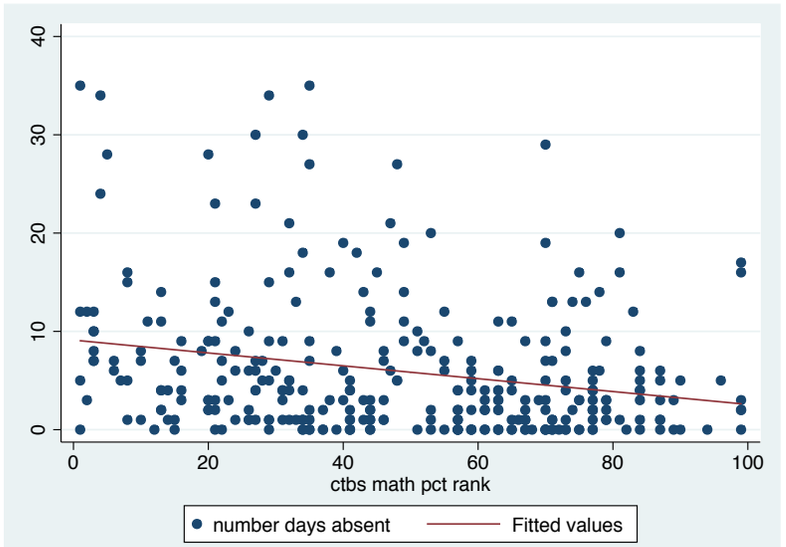


(e) Original scale for absence

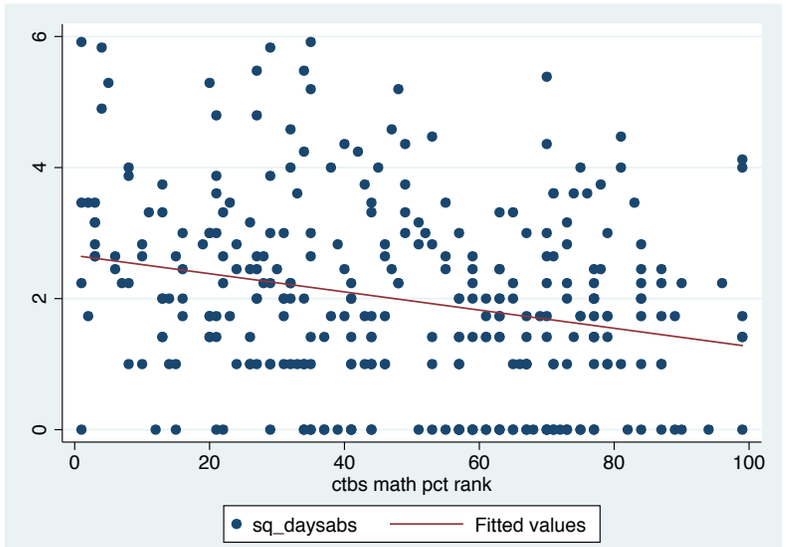


(f) square root scale for absence

Regression line



(g) Original scale for absence

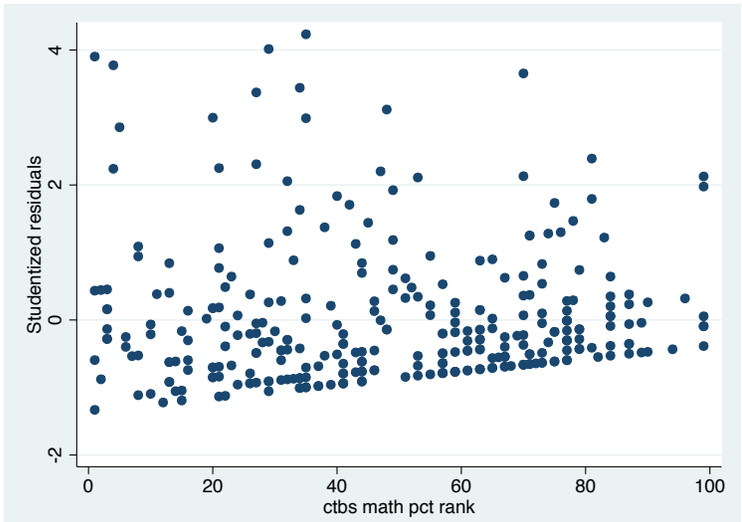


(h) square root scale for absence

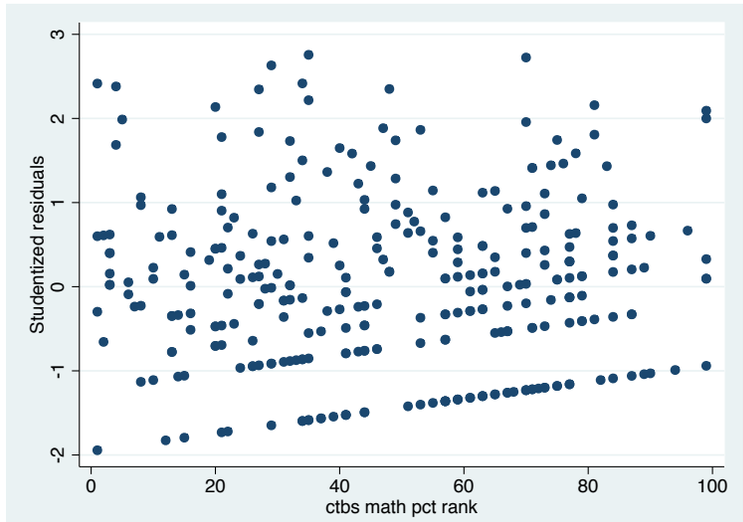
Apparent decreasing variation in days missed as math scores increase for the untransformed counts, less so for transformed values.

```
. regress daysabs math
. predict resid1, rstudent
. regress sq_daysabs math
```

```
. predict resid2, rstudent
. scatter resid1 math
. scatter resid2 math
```



(i) Original scale for absence



(j) square root scale scale for absence

The residual plot and fit look better under the transform. This type of data might be better addressed with a different type of model (later in course), but here the use of linear regression is (more) justified after addressing the variance issue

Power Transformations

SPRM Section 7.18

Transforming via some multiplicative scaling (powers, logs) works works well in many cases. Recall that we can identify non-linearities mostly through simple plotting, such as scatterplots Residuals plots work well for this purpose, especially in multiple regression.

A general approach that includes log and power transformations is the Box-Cox transformation (Box and Cox 1964 *J Royal Stat Soc*)

Box-Cox Transformation: Transform Y into Y' via:

$$Y' = (Y^{\lambda} - 1)/\lambda$$

We can use maximum likelihood estimation to find (and test hypotheses about) λ. This is facilitated in Stata by the "boxcox" command, and in R by a similar "boxcox" module

Box-Cox Transformation

- This approach may allows us to choose and/or simplify the transform, based on the data
- Essentially, finds what transform of Y produces the best model for a given covariate set. Can then also test specific hypotheses about the value of λ

- **If, after obtaining the λ estimate, the hypothesis $H_0 : \lambda = 0$ cannot be rejected**, simply use $Y' = \log_e(Y)$ instead of the estimated λ would work, because it can be shown that:

$$\text{as } \lambda \to 0, \text{ the transform } Y' \to \log_e(Y)$$

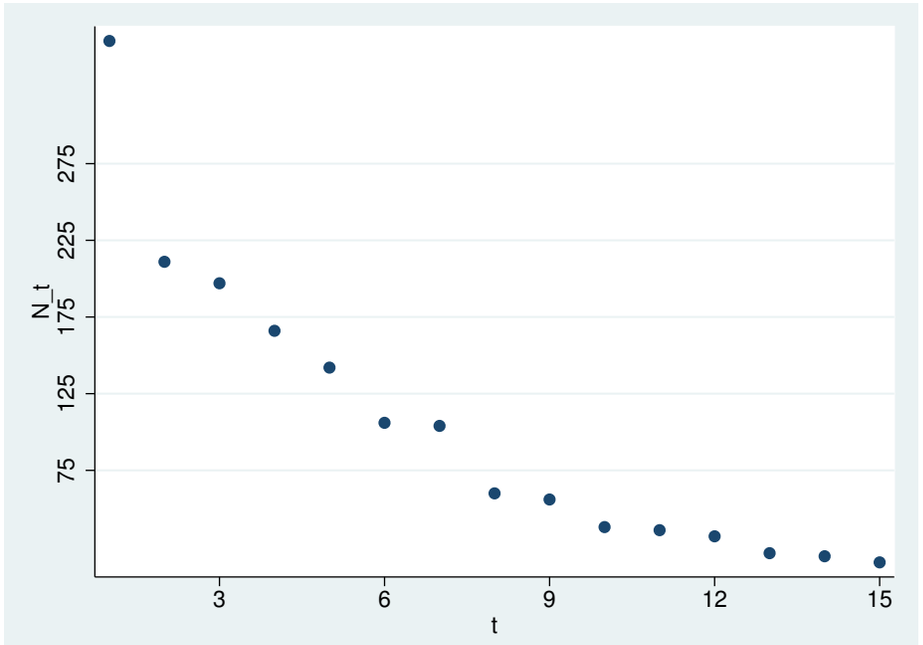
- Similarly, **if $H_0 : \lambda = -1$ cannot be rejected**, use $Y' = 1/Y$ (the reciprocal of Y) would work.
- **If $H_0 : \lambda = 1$ cannot be rejected** (and data look ok), use $Y' = Y$ (i.e., do nothing, use Y directly)

Box-Cox Transformations

- **Some properties of the Box-Cox transformation**
  - **It is rank-order preserving** - does not re-arrange values according to original magnitude
  - contains common transformations:
    * when $\lambda = -1$, we have inverse transform
    * When $\lambda \to 0$, we have natural log transform
  - All other 'ladders of power' are represented (square root, square, etc)

Box-Cox Transformations

- **Ex** from C&H Table 6.2: Number of surviving bacteria (Units of 100) estimated by plate counts following exposure to 200-kilovolt X-rays for periods ranging from $t = 1$ to 15 intervals of 6 minutes. The scatter plot of number surviving vs. time exposed:



- It is common for 'kill' proportions, etc in biological systems to follow some sort of power law

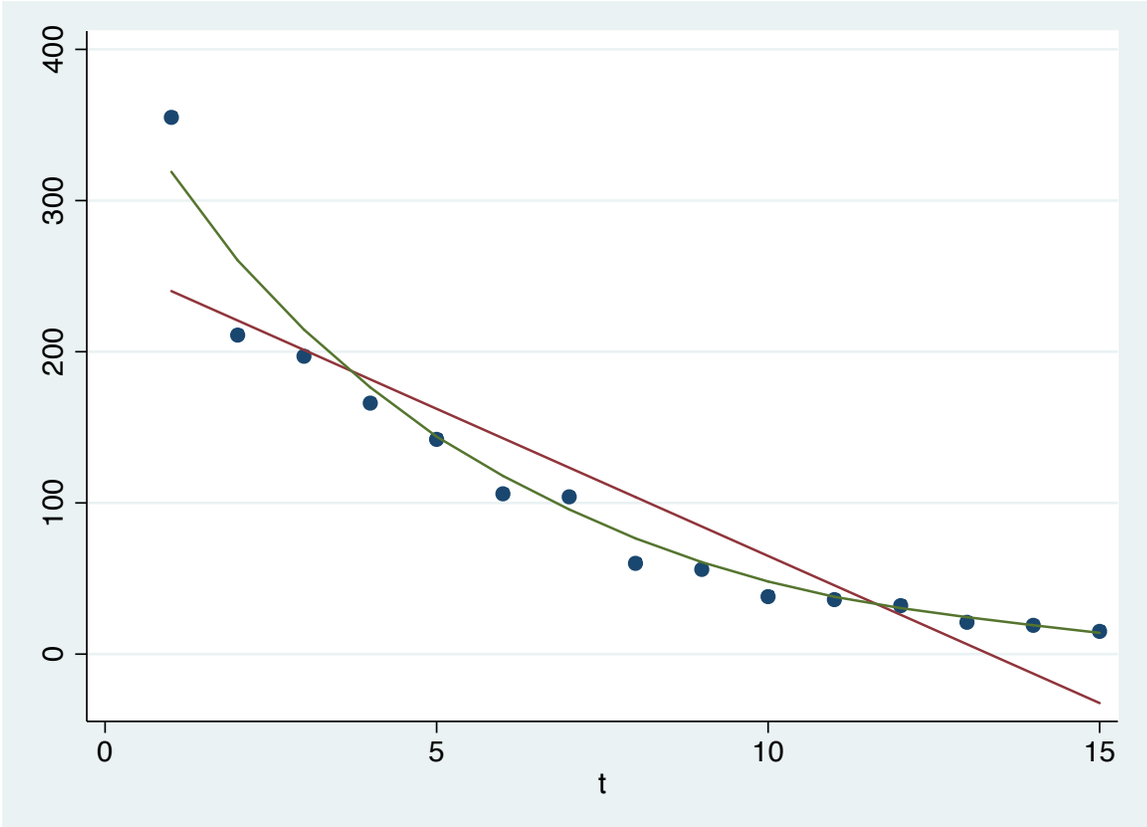
We model number of bacteria by time exposed.

```
. regress n_t   t

      Source |       SS       df       MS              Number of obs =      15
-------------+------------------------------           F(  1,    13) =   60.62
       Model |  106080.357     1  106080.357            Prob > F      =  0.0000
    Residual |  22749.3762    13  1749.95201            R-squared     =  0.8234
-------------+------------------------------           Adj R-squared =  0.8098
       Total |  128829.733    14  9202.12381            Root MSE      =  41.832

------------------------------------------------------------------------------
         n_t |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           t |  -19.46429   2.499966    -7.79   0.000    -24.86513   -14.06344
       _cons |    259.581   22.72999    11.42   0.000     210.4758    308.6861
------------------------------------------------------------------------------
```

Box-Cox Transformations

We can also plot with 'smoothed' curve to get some sense of deviation from linearity.

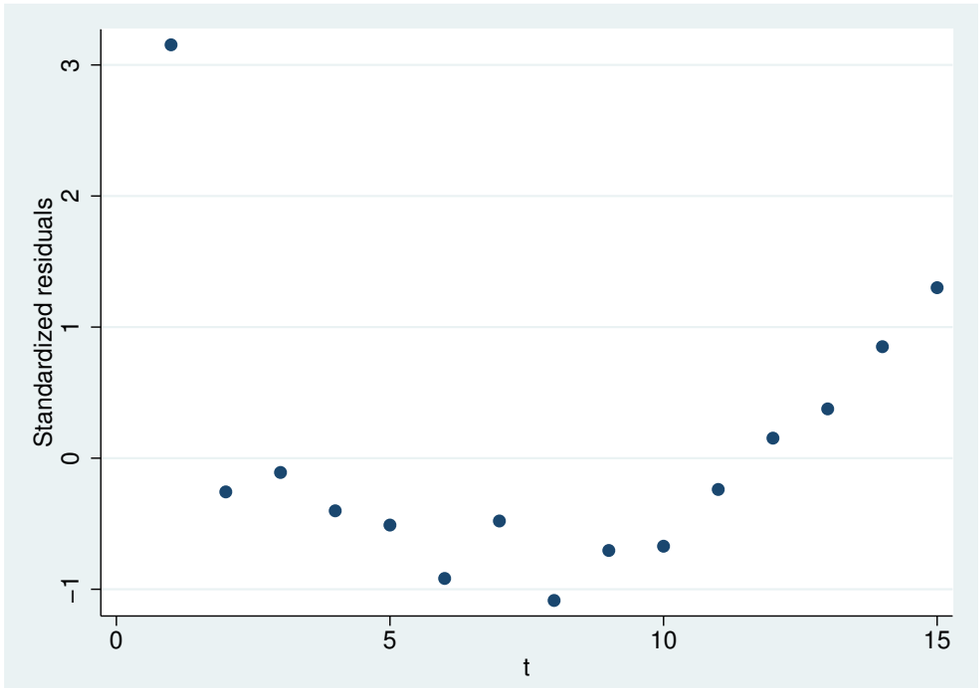
```
twoway (scatter n_t t) (lfit n_t t) (lowess n_t t), legend(off)
```



Transformations

Check the residual plot for the linear fit.

```
. predict sres, rstandard
. scatter sres t
```



In addition to the pattern in the residual related to t, one observation with very large residual is associated with the smallest t value. Let's do Box-Cox analysis.

Box-Cox Analysis

- This procedure estimates the value of λ by searching for the value that minimizes the sum of squares of residuals in the model

$$Y_i^{(\lambda)} = \beta_0 + \beta_1 + \ldots + \beta_p + \epsilon_i$$

```
. . boxcox n_t t
Fitting comparison model

Iteration 0:   log likelihood = -89.220553
Iteration 1:   log likelihood = -84.049995
. . .

Fitting full model

Iteration 0:   log likelihood = -76.215899  (not concave)
Iteration 1:   log likelihood = -51.311636
. . .
                                                Number of obs   =         15
                                                LR chi2(1)      =      66.89
Log likelihood = -50.444818                     Prob > chi2     =      0.000
```

```
------------------------------------------------------------------------------
         n_t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      /theta |  -.0195753   .0648972    -0.30   0.763    -.1467715     .107621
------------------------------------------------------------------------------
```

- **This value** (theta) is the estimated value of the power parameter (i.e., the λ). Note that is is small (close to zero)

```
Estimates of scale-variant parameters
----------------------------
             |      Coef.
-------------+--------------
Notrans      |
           t |   -.201097
       _cons |   5.656359
-------------+--------------
      /sigma |   .0940029
----------------------------
```

- **These are the β coefficients that would be obtained were this transformation applied and then an ordinary regression run.**

That is, the model is

$$\hat{n}'(t) = 5.656359 - 0.201 * t$$

where

$$n(t)' = (n(t)^{-.0195753} - 1)/ - .0195753$$

Again, the steps that are carried out are:

1. finding the λ that minimizes the residual error in $Y_i^{(\lambda)} = \beta_0 + \beta_1 + \ldots + \beta_p + \epsilon_i$
2. Use that λ obtained to transform Y into a new Y' via $Y' = (Y^{\lambda} - 1)/\lambda$
3. Do the ordinary least squares using Y' as the response

Box-Cox Analysis

Run this analysis:

```
. gen N_t_mod = (n_t^(-.0195753)-1)/(-.0195753)
. reg N_t_mod t

      Source |       SS           df       MS      Number of obs   =        15
-------------+----------------------------------   F(1, 13)        =   1110.55
       Model |  11.3232013         1  11.3232013   Prob > F        =    0.0000
    Residual |  .132548371        13  .010196029   R-squared       =    0.9884
-------------+----------------------------------   Adj R-squared   =    0.9875
       Total |  11.4557497        14  .818267834   Root MSE        =    .10098

------------------------------------------------------------------------------
     N_t_mod | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
           t |   -.201097   .0060344   -33.32   0.000    -.2141336   -.1880604
       _cons |   5.656358   .0548658   103.09   0.000     5.537828    5.774888
------------------------------------------------------------------------------
```

Looks much better. Note that the β slope is what the Box-Cox analysis indicated, but now we have test, CI, etc for this parameter.

Box-Cox Analysis

The procedure provides a test of various values for λ, which relate to specific common transformations

```
-------------------------------------------------------
   Test          Restricted     LR statistic      P-value
    H0:       log likelihood       chi2       Prob > chi2
-------------------------------------------------------
theta = -1       -74.02278        47.16           0.000
theta =  0      -50.490254         0.09           0.763
theta =  1      -76.215899        51.54           0.000
-------------------------------------------------------
```

- **These are hypothesis tests on the λ parameter for the transformation**. Note that value 0 cannot be rejected. This suggests a log transformation may be appropriate (recall that as λ tends towards zero, $Y' \approx \log_e(Y)$)

Box-Cox Analysis

Since the analysis indicated log transform may work, we try the log transform for the # of surviving bacteria, and re-run.

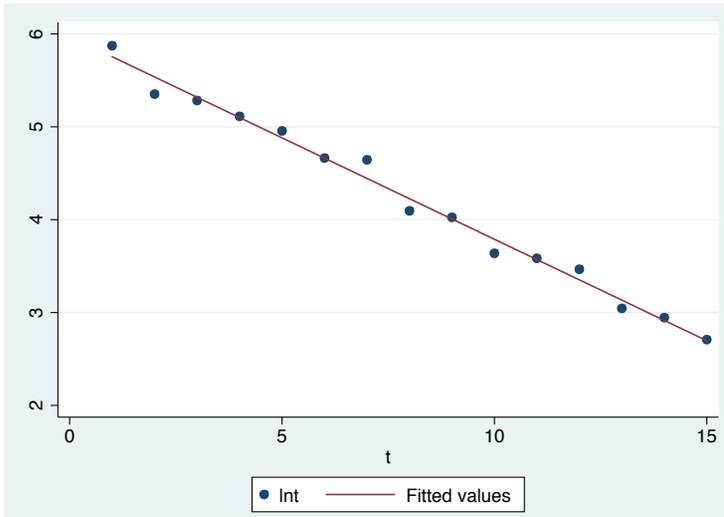
```
. generate lnt = log(n_t)
. reg lnt t
      Source |       SS       df       MS              Number of obs =      15
-------------+------------------------------           F(  1,    13) = 1103.70
       Model |  13.3586861     1  13.3586861            Prob > F      =  0.0000
    Residual |  .157345913    13  .012103532           R-squared     =  0.9884
-------------+------------------------------           Adj R-squared =  0.9875
       Total |   13.516032    14  .965430858           Root MSE      =  .11002
------------------------------------------------------------------------------
         lnt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           t |  -.2184253   .0065747   -33.22   0.000    -.2326291   -.2042214
       _cons |    5.97316   .0597781    99.92   0.000     5.844018    6.102303
------------------------------------------------------------------------------
```

- Note that the above β estimates are *very close* to those of the suggested Box-Cox power transform ($\beta_1 = -.201$, $\beta_0 = 5.66$). Also, R^2 is greatly improved.

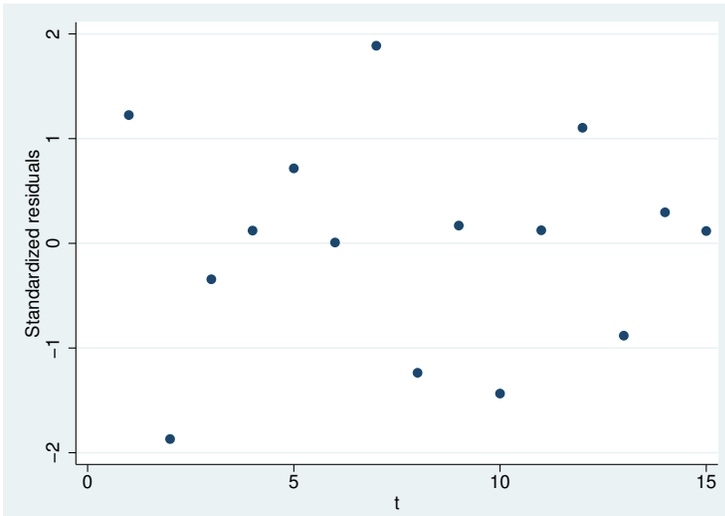
Analysis - log-transformed data

```
. predict  lsres, rstandard
. scatter lsres t
. scatter lnt t, ylabel(3(.75)5.25) xlabel(3(3)15)
```

Now the residual plot and fit look much better.



(k) Regression line



(l) Residuals

Transformations for Both Response and Predictor

- Sometimes, transformations are needed for both Y and X, to deal with model violations and/or scaling issues.

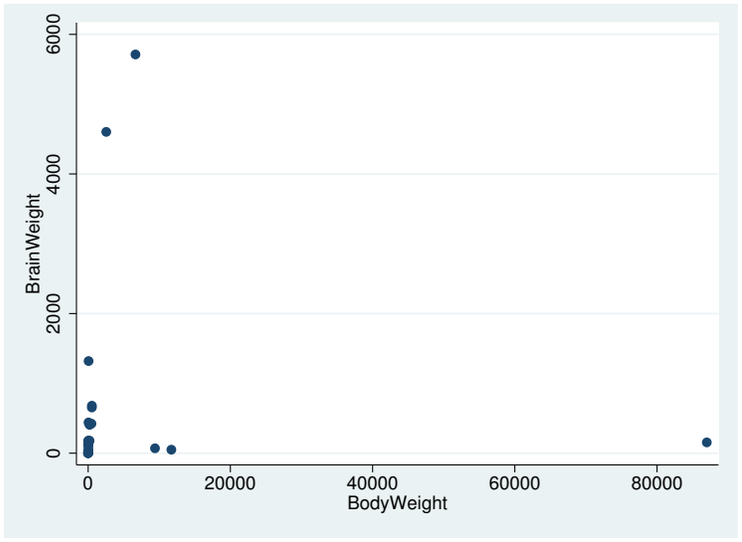
  Example: Data on brain weight (grams) vs. body weight (kilograms) in mammals (and some large proto-reptiles) (Table 6.14 pf C&H)

```
. use http://www.ats.ucla.edu/stat/stata/examples/chp/p176, clear
. list name brainwei bodyweig, clean
```

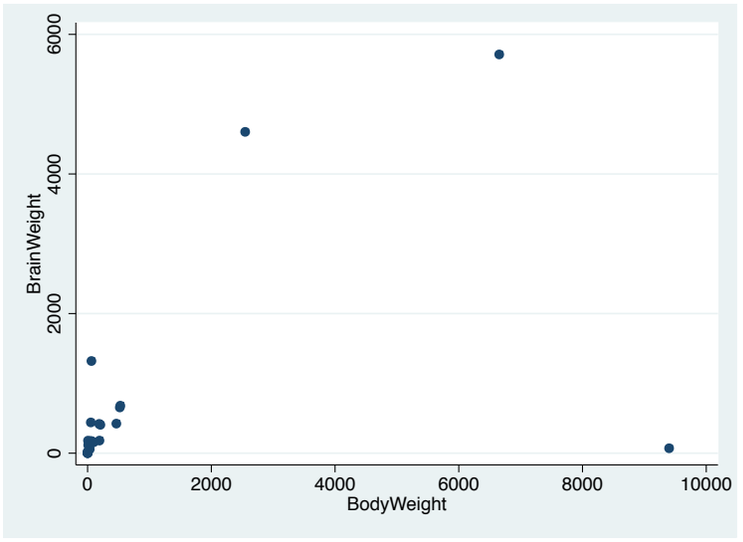
	name	brainwei	bodyweig
1.	Mountain beaver	8.1	1.35
2.	Cow	423	465
3.	Graywolf	119.5	36.33
4.	Goat	115	27.66
5.	Guineapig	5.5	1.04
6.	Diplodocus	50	11700
7.	Asian elephant	4603	2547
8.	Donkey	419	187.1
9.	Horse	655	521
10.	Potar monkey	115	10
11.	Cat	25.6	3.3
12.	Giraffe	680	529

13.	Gorilla	406	207
14.	Human	1320	62
15.	African elephant	5712	6654
16.	Triceratops	70	9400
17.	Rhesus monkey	179	6.8
18.	Kangaroo	56	35
19.	Hamster	1	.12
20.	Mouse	.4	.023
21.	Rabbit	12.1	2.5
22.	Sheep	175	55.5
23.	Jaguar	157	100
24.	Chimpanzee	440	52.16
25.	Brachiosaurus	154.5	87000
26.	Rat	1.9	.28
27.	Mole	3	.122
28.	Pig	180	192

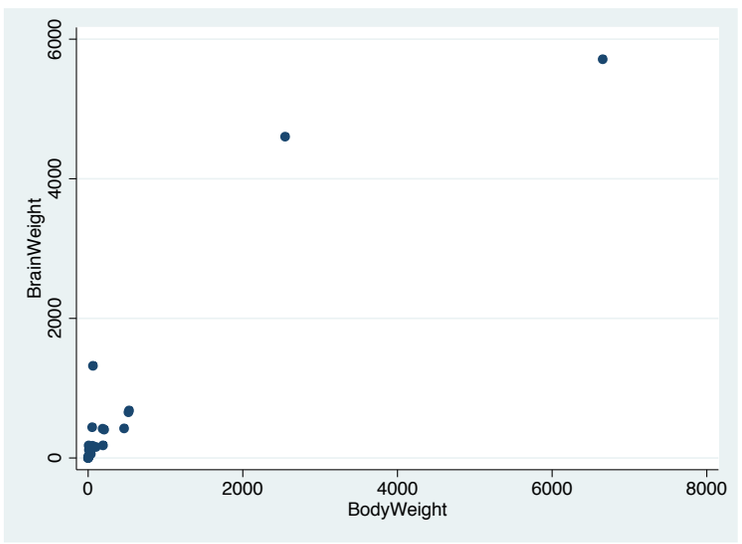
This data has very large range for the body weight variable, large range for brain weight also. We make some scatter plots.



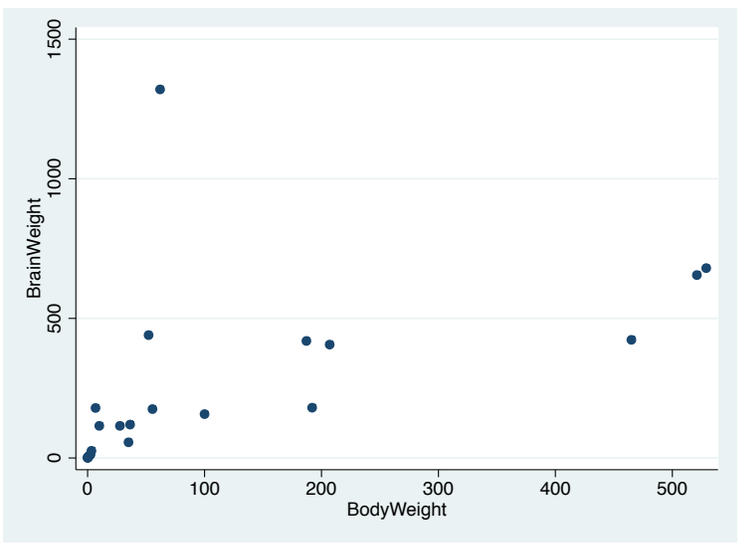
(m) All data



(n) Removing brachiosaurus, diplodicus



(o) Removing all dinosaurs



(p) Removing elephants

Transformations for Both Response and Predictor

- There is no correlation (on this scale) on full dataset, moderate correlation for the smaller dataset

```
. corr brainwei bodyweig
(obs=28)

             | brainwei bodyweig
-------------+------------------
    brainwei |   1.0000
    bodyweig |  -0.0053   1.0000
```

```
.* remove dinos and elephants
. corr brainwei bodyweig if bodyweig < 2500
(obs=23)

             | brainwei bodyweig
-------------+------------------
    brainwei |   1.0000
    bodyweig |   0.5424   1.0000
```

However. we want to analyze the entire dataset. Will it work on some other scale?

Transformations for Both Response and Predictor

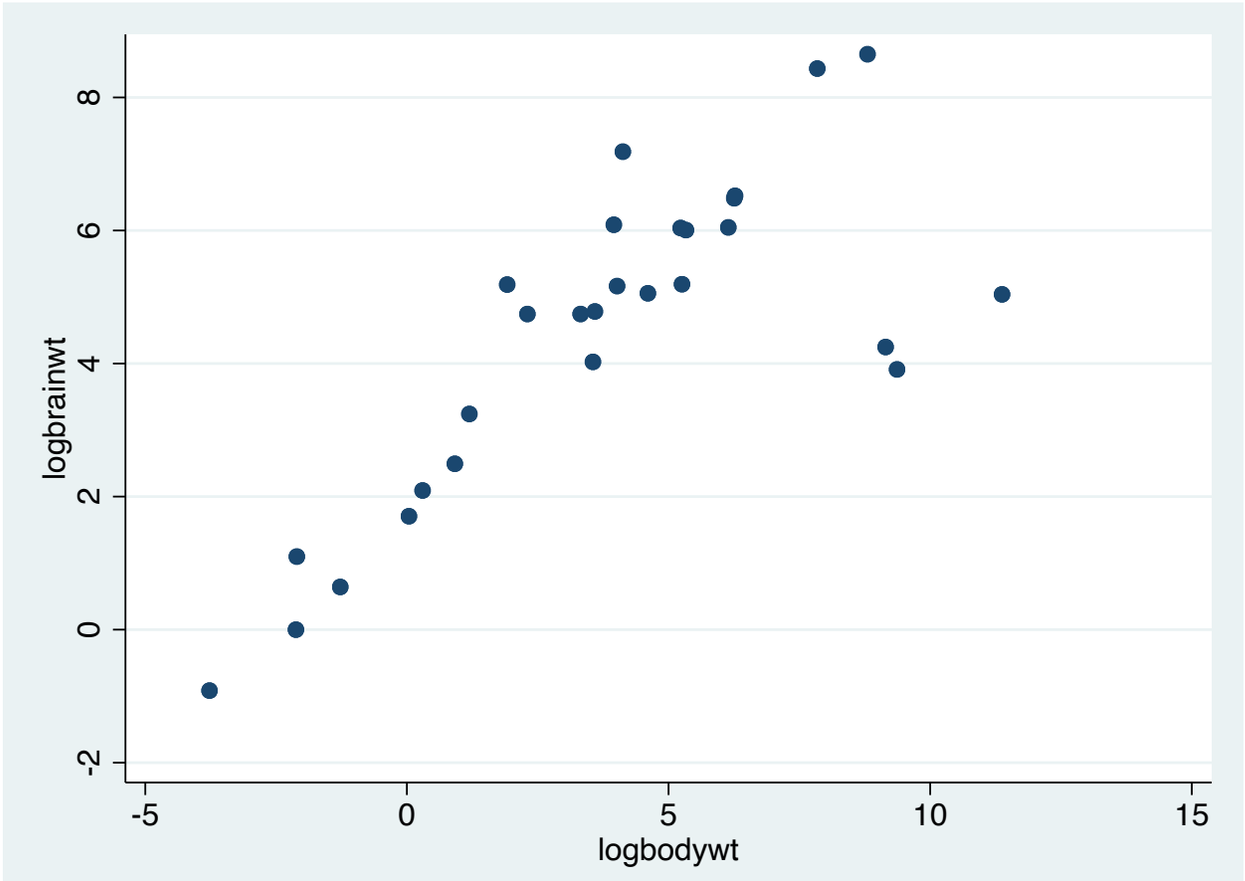
- We can consider 'ladders of power' (Y^λ, X^λ for $\lambda = -1, -.5, 0, 2, \ldots$) to see which provides a more linear appearing plot.

- We could also try the Box-Cox approach, although the range of data is so extreme that some estimation problems result.

- Once again, it turns out that **log transforms, this time on on both scales, looks best**. This is because there are some natural 'outliers' that, if removed, would improve the fit substantially

Transformations for Both Response and Predictor

- Plot with natural log of both brain weight and body weight (all data):



Transformations for Both Response and Predictor

```
      Source |       SS       df       MS              Number of obs =      28
-------------+------------------------------           F(  1,    26) =   40.26
       Model |  94.4390272     1  94.4390272            Prob > F      =  0.0000
    Residual |  60.9879893    26   2.3456919            R-squared     =  0.6076
-------------+------------------------------           Adj R-squared =  0.5925
       Total |  155.427017    27  5.75655617            Root MSE      =  1.5316

------------------------------------------------------------------------------
     logbrwt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     logbody |   .4959947   .0781694     6.35   0.000     .3353152    .6566742
       _cons |   2.554898    .413137     6.18   0.000     1.705683    3.404113
-------------+----------------------------------------------------------------
```

- Fit is much improved, with ln(body wt) as significant predictor of ln(brain wt)

- Interestingly, on the original scale, there is no detectable relationship, due in part to huge variance in both variables

```
regress brainwei bodyweig
```

```
      Source |       SS       df       MS              Number of obs =      28
-------------+------------------------------           F(  1,    26) =    0.00
       Model |  1372.62473     1  1372.62473            Prob > F      =  0.9785
    Residual |  48113597.9    26  1850522.99            R-squared     =  0.0000
-------------+------------------------------           Adj R-squared = -0.0384
       Total |  48114970.5    27  1782035.94            Root MSE      =  1360.3

------------------------------------------------------------------------------
    brainwei |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    bodyweig |  -.0004326   .0158853    -0.03   0.978    -.0330853    .0322201
       _cons |   576.3724   265.9121     2.17   0.040     29.78228    1122.963
------------------------------------------------------------------------------
```

One More Example with Box-Cox and a Simple Transform

Recall one of the Power Transformations is Y^{-1} or an inverse transform (i.e., equals $1/Y$)

- Order preserving, reverses the order
- bounded at zero, only valid for nonzero Y
- like logarithms, tends to reign in extreme values

One More Example with Box-Cox and a Simple Transform

Revisiting the Port wine example:

```
.  boxcox costd age
Fitting comparison model

Iteration 0:   log likelihood = -84.948369
Iteration 1:   log likelihood = -71.009893
Iteration 2:   log likelihood = -71.009838
Iteration 3:   log likelihood = -71.009838

Fitting full model

Iteration 0:   log likelihood =  -69.59725  (not concave)
. . .
Iteration 4:   log likelihood = -43.865893

                                                Number of obs   =         19
                                                LR chi2(1)      =      54.29
Log likelihood = -43.865893                     Prob > chi2     =      0.000
------------------------------------------------------------------------------
       costd |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      /theta |  -.5723598   .1958182    -2.92   0.003    -.9561565   -.1885631
```

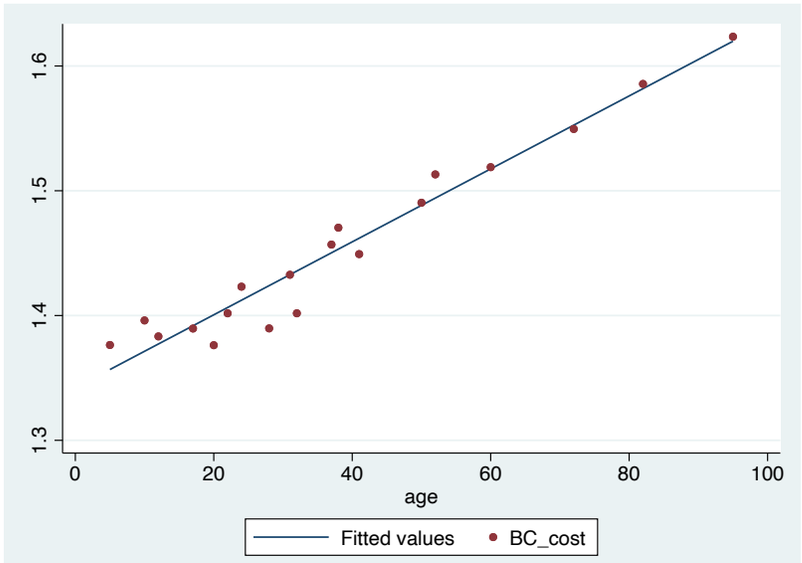
```
------------------------------------------------------------------------------
Estimates of scale-variant parameters
----------------------------
             |      Coef.
-------------+--------------
Notrans      |
         age |   .0029229
       _cons |   1.342042
-------------+--------------
      /sigma |   .0162165
----------------------------

-----------------------------------------------------------
   Test          Restricted     LR statistic      P-value
    H0:       log likelihood       chi2       Prob > chi2
-----------------------------------------------------------
theta = -1      -45.599771         3.47           0.063
theta =  0      -48.760615         9.79           0.002
theta =  1       -69.59725        51.46           0.000
-----------------------------------------------------------
```

Based on the result, we can use $\lambda = -.5723$, OR because we can't reject $H_0 : \lambda = -1$, we can consider inverse transformation also.

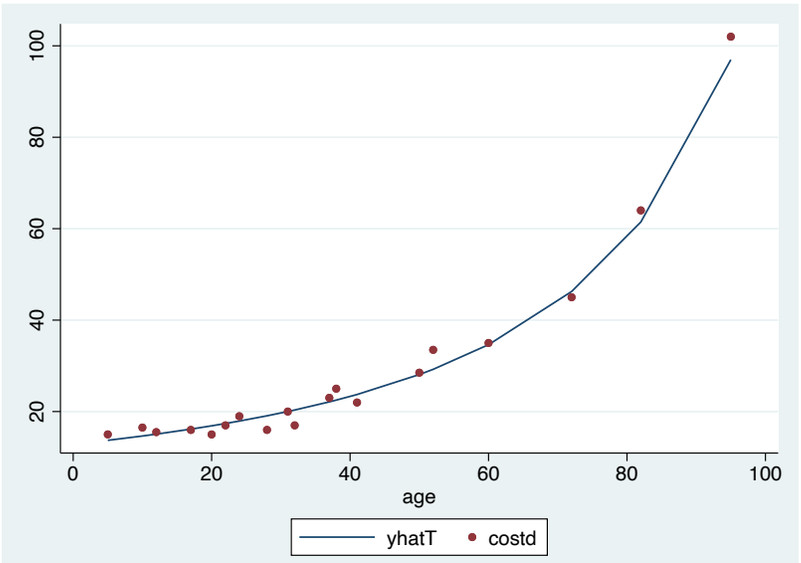
Calculate the actual transform via λ and check the fit

```
. gen BC_cost = (costd^(-0.5723)-1)/(-.5723)

. reg BC_cost age

      Source |       SS           df       MS      Number of obs   =        19
-------------+----------------------------------   F(1, 17)        =    323.37
       Model |  .095077723         1  .095077723   Prob > F        =    0.0000
    Residual |  .004998423        17  .000294025   R-squared       =    0.9501
-------------+----------------------------------   Adj R-squared   =    0.9471
       Total |  .100076146        18  .005559786   Root MSE        =    .01715

------------------------------------------------------------------------------
     BC_cost |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .0029235   .0001626    17.98   0.000     .0025805    .0032665
       _cons |   1.342117   .0073674   182.17   0.000     1.326573    1.357661
------------------------------------------------------------------------------

. predict yhat
(option xb assumed; fitted values)
```

```
.  scatter yhat BC_cost age , c(l .) s(i o)

. * transform yhat back
.  gen yhatT=(yhat*-0.5723+1)^(1/-0.5723)

. sc pbhatT price age, c(l .) s(i o)
```



(q) Box-Cox, $\lambda = -0.5723$, price$'$ vs age



(r) original scale, price vs age

But what if we just did inverse transform?

```
. gen invcost = 1/costd

. reg invcost age

      Source |       SS           df       MS      Number of obs   =        19
-------------+----------------------------------   F(1, 17)        =    237.48
       Model |  .005485528         1  .005485528   Prob > F        =    0.0000
    Residual |  .000392685        17  .000023099   R-squared       =    0.9332
-------------+----------------------------------   Adj R-squared   =    0.9293
       Total |  .005878213        18  .000326567   Root MSE        =    .00481

------------------------------------------------------------------------------
     invcost |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  -.0007022   .0000456   -15.41   0.000    -.0007984   -.0006061
       _cons |   .0729182    .002065    35.31   0.000     .0685614    .077275
```

Looks quite good, as good as log transform used earlier. Fit on extreme points appears better based on plot

More Transformations
Transformations for Percentages

- For response Y is in the form of a proportion (range [0,1]) or percentage (proportion X 100), we may wish to treat it is a random variable coming from a continuous distribution. Several concerns:

  1. Model resulting may not predict a proportion or percentage (will not be naturally constrained to do so). This may be ok depending on range of Y and X values
  2. Distribution may be highly skewed, not Normal (Gaussian) - can possibly fix
  3. For proportions, variance depends on the mean - a violation of regression assumptions - can possibly fix

Transformations for Percentages

- **Important note** Here, we are taking about Y recorded as a proportion of percentage of some n, where each individual n_i is a yes/no, 0/1 type outcome.

    - For example, proportion of cells killed at some dose of a cytotoxic agent, or proportion of hospitals in a region having an open MRI machine.

    - We are not talking about situations where the observations are individuals responses (0/1, yes/no, etc). This will handled later (SPRM Ch. 11, 12)

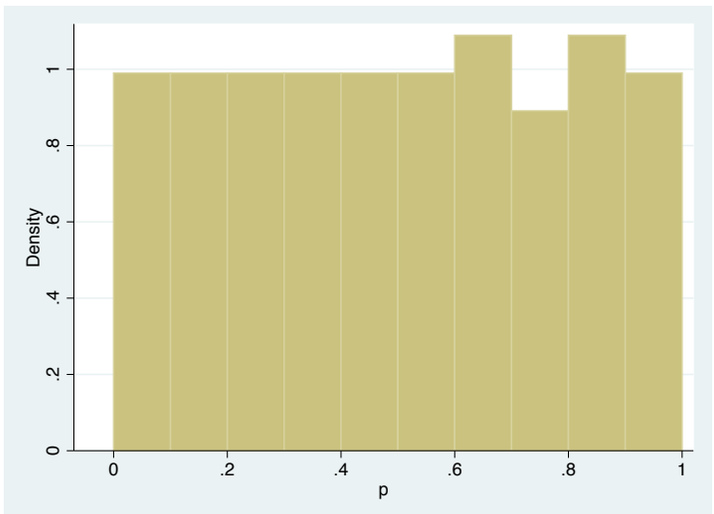
- To see how different transformations may help in this data situation, we generate some hypothetical data

Transformations for Percentages

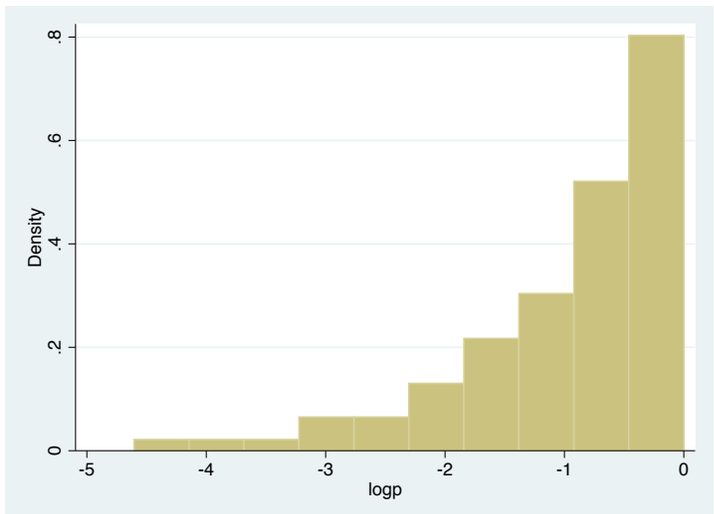
```
. set obs 101
. range p 0 1
. gen logp = log(p)
. * arcsin squareroot transform is frequently recommended (see pg 173)
. gen arcsin = asin(sqrt(p))
. * logic transform log(p/1-p) is useful
. gen logit   = logit(p)
```

- Make some plots and summary statistics

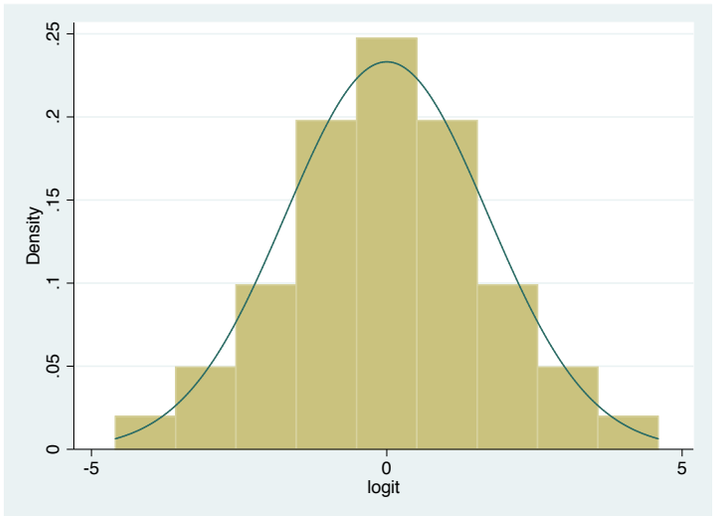
```
 hist p
(bin=10, start=0, width=.1)
. hist logp
(bin=10, start=-4.6051702, width=.46051702)
. histogram logit, normal
(bin=9, start=-4.59512, width=1.0211379)
. histogram arcsin, normal
(bin=10, start=0, width=.15707964)

. summarize p, detail
. summarize logp, detail
. summarize logit, detail
. summarize arcsin, detail
```



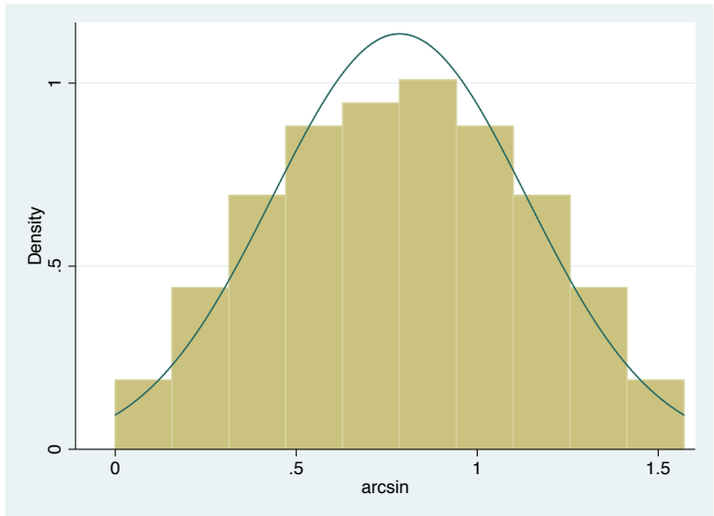
(s) raw p



(t) log(p)



(u) logit transform



(v) arcsin square-root transform

Transformations for Percentages

- Where this type of data often appears (ecology, biology, for example), there are guidelines and 'rules of thumb' for what should be used. square-root arcsin transformation works fairly well.

- There seems to be a movement towards using the logit transform for this type of data, which seems to have better behavior in terms of correcting residuals. Logit transform is

$$\text{logit}(p) = \log \frac{p}{1-p}$$

- An alternate solution is to abandon linear regression for the problem, we will discuss later

There is flexibility in choice of transform, but it must make sense mathematically

Transformations - Summary

- Transformations are a useful tool to adapt data to better fit linear models assumptions and requirements
- Not all problems can be fixed, should be used judiciously, and other models may be more appropriate
- Interpretation of analysis results on transformed data must take into account the transformed scale, or be transformed back for clearer presentation