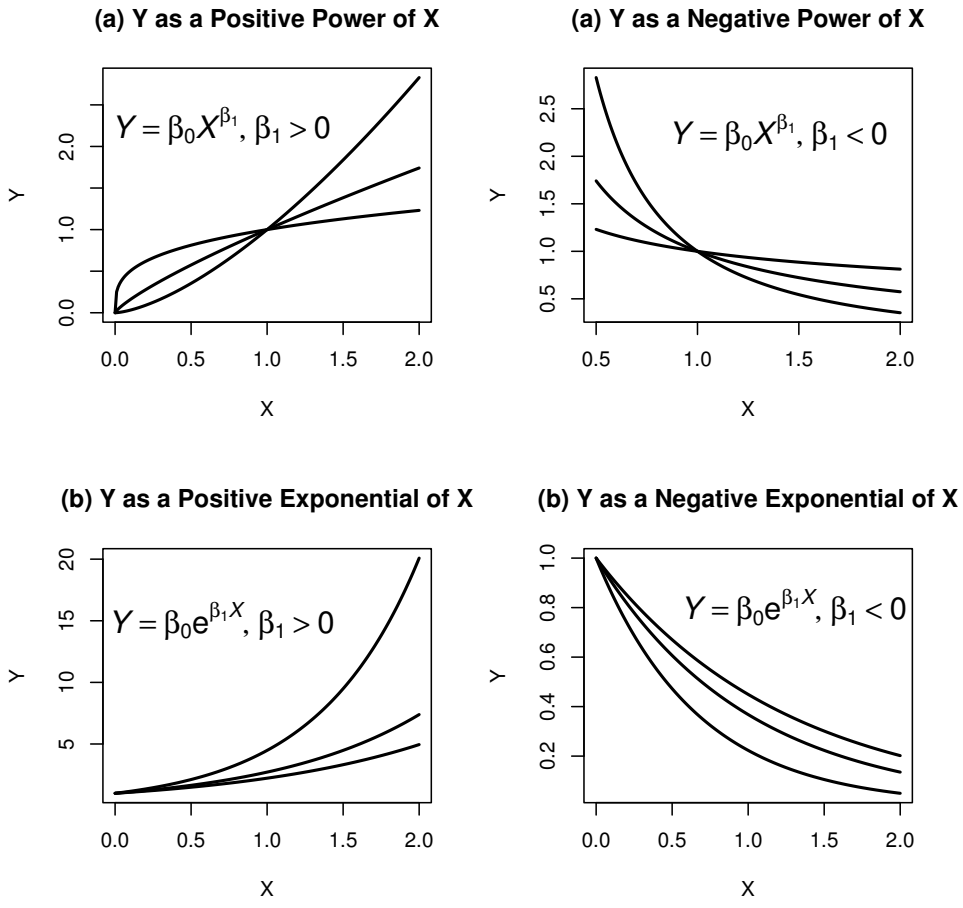
3 Transformations in Regression

Simple linear regression is appropriate when the scatterplot of Y against X show a linear trend. In many problems, non-linear relationships are evident in data plots. Linear regression techniques can still be used to model the dependence between Y and X, provided the data can be transformed to a scale where the relationship is roughly linear. In the ideal world, theory will suggest an appropriate transformation. In the absence of theory one usually resorts to empirical model building. Polynomial models are another method for handling nonlinear relationships.

I will suggest transformations that you can try if the **trend** in your scatterplot has one of the following functional forms. The responses are assumed to be non-negative (in some cases strictly positive) in all cases.



The functional relationship between Y and X in (a) is given by $Y = \beta_0 X^{\beta_1}$, that is Y is related to a power of X, where the power is typically unknown. For the left plot, $\beta_1 > 0$ whereas $\beta_1 < 0$ for the plot on the right. For either situation, the logarithm of Y is linearly related to the logarithm of X (regardless of the base):

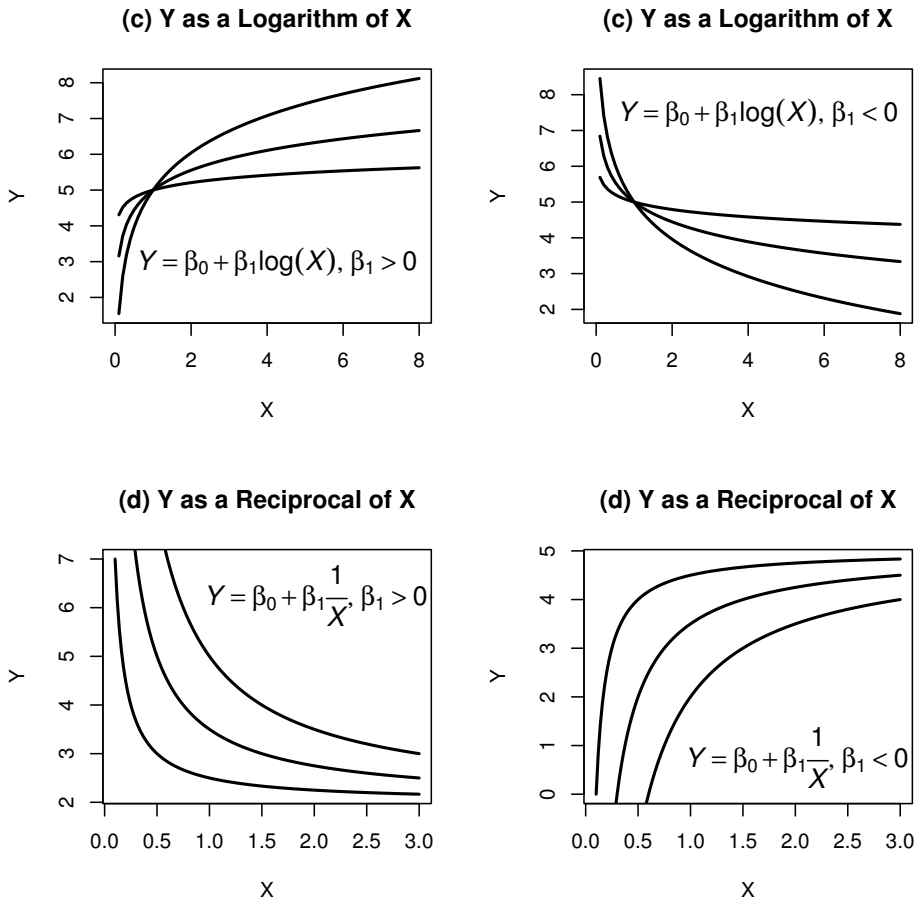
$$\log(Y) = \log(\beta_0) + \beta_1 \log(X).$$

You should consider a simple linear regression of $Y' = \log(Y)$ on $X' = \log(X)$.

The functional relationship between Y and X in (b) is given by $Y = \beta_0 \exp(\beta_1 X)$, that is Y is an exponential function of X. For the plot on the left, $\beta_1 > 0$ whereas $\beta_1 < 0$ for the plot on the right. In either situation, the natural logarithm of Y is linearly related to X:

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

You should consider a simple linear regression of $Y' = \log_e(Y)$ on X. Actually, the base of the logarithm is not important here either.



The functional relationship between Y and X in (c) is given by $Y = \beta_0 + \beta_1 \log(X)$, that is Y is an logarithmic function of X. For the plot on the left, $\beta_1 > 0$ whereas $\beta_1 < 0$ for the plot on the right. In each situation, consider a simple linear regression of Y on $X' = \log(X)$.

The functional relationship between Y and X in (d) is

$$Y = \beta_0 + \beta_1 \frac{1}{X}.$$

Hence, consider a simple linear regression of Y on $X' = 1/X$. Note that each plot in (d) has a horizontal asymptote of β_0.

In most problems, the trend or signal will be buried in a considerable amount of noise, or variability, so the best transformation may not be apparent. If two or more transformations are suggested try all of them and see which is best - look at diagnostics from the various fits rather than (meaningless) summaries such as R^2. In situations where a logarithmic transformation is suggested, you might try a square root transformation as well. It often does make a considerable difference in the quality of the fit whether you transform Y only, X only, or both. There are more organized schemes for choosing transformations, but this sort of trial and error is the most common practice. Note that the functional forms (a) - (d), while probably the most frequently encountered, are not at all the only ones used.

The need to transform is sometimes much more apparent in a plot of the residuals against the predicted values from a "linear fit" of the **original data** because you tend not to perceive subtle deviations from linearity. The *Wind Speed* example below illustrates this.

Transformations also can help to control influential values and outliers (recall that an outlying X-value can cause that point to exert undue influence on the fit). Functions such as *log* have the effect of bringing outlying values much closer to the rest of the data. The *Brain Weights vs. Body Weights* example below illustrates this. When I see a variable with a highly skewed distribution, I usually try transforming it to make it more symmetric. This can work both ways, of course - you can make a nice symmetrically distributed variable skewed by transforming it.

Computing Predictions

Transforming the response to a new scale causes no difficulties if you wish to make predictions on the original scale. For example, suppose you fit a linear regression of $\log_e(Y)$ on X. The fitted values satisfy

$$\widehat{\log_e(Y)} = b_0 + b_1 X.$$

The predicted response Y_p for an individual with $X = X_p$ is obtained by first getting the predicted value for $\log_e(Y_p)$:

$$\widehat{\log_e(Y_p)} = b_0 + b_1 X_p.$$

Our best guess for Y_p is obtained by exponentiating our prediction for $\log_e(Y_p)$:

$$\hat{Y}_p = \exp(\widehat{\log_e(Y_p)}) = \exp(b_0 + b_1 X_p).$$

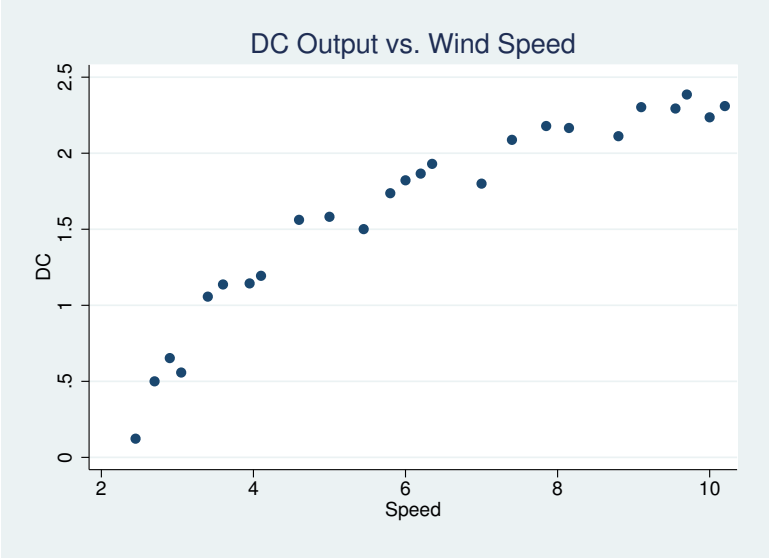
The same idea can be used to get prediction intervals for Y_p from a prediction interval for $\log_e(Y_p)$ (just transform the lower and upper confidence limits).

Other transformations on Y are handled analogously. For example, how do you predict Y using a simple linear regression with $1/Y$ as the selected response?

Example of Transformations: Wind Speed Data

A research engineer is investigating the use of a windmill to generate electricity. She has collected data on the DC output from the windmill and the corresponding wind velocity. She wants to develop a model that explains the dependence of the DC output on wind velocity. The data were read into **Stata** and plotted.

```
. list speed dc,clean
       speed      dc
  1.       5   1.582
  2.       6   1.822
  3.     3.4   1.057
  4.     2.7      .5
  5.      10   2.236
  6.     9.7   2.386
  7.    9.55   2.294
  8.    3.05    .558
  9.    8.15   2.166
 10.     6.2   1.866
 11.     2.9    .653
 12.    6.35    1.93
 13.     4.6   1.562
 14.     5.8   1.737
 15.     7.4   2.088
 16.     3.6   1.137
 17.    7.85   2.179
 18.     8.8   2.112
 19.       7     1.8
 20.    5.45   1.501
 21.     9.1   2.303
 22.    10.2    2.31
 23.     4.1   1.194
 24.    3.95   1.144
 25.    2.45    .123
```

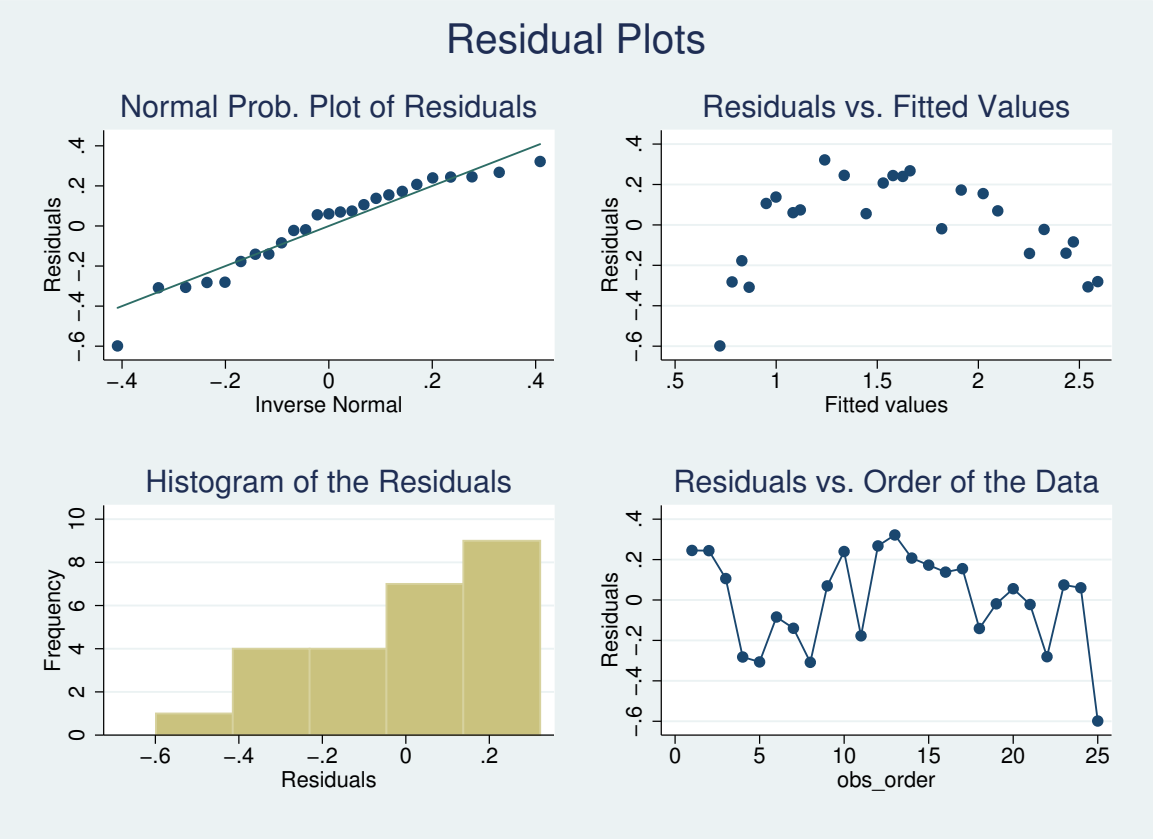



```
. regress dc speed
      Source |       SS       df       MS              Number of obs =      25
-------------+------------------------------           F(  1,    23) =  160.26
       Model |  8.92961408     1  8.92961408           Prob > F      =  0.0000
    Residual |  1.28157328    23  .055720577           R-squared     =  0.8745
-------------+------------------------------           Adj R-squared =  0.8690
       Total |  10.2111874    24   .42546614           Root MSE      =  .23605
------------------------------------------------------------------------------
          dc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       speed |   .2411489   .0190492    12.66   0.000     .2017426    .2805551
       _cons |   .1308752   .1259894     1.04   0.310    -.1297537    .3915041
------------------------------------------------------------------------------
```

The data plot shows a strong linear trend, but the relationship is nonlinear. If I ignore the nonlinearity and fit a simple linear regression model, I get

$$\text{Predicted DC Output} = .1309 + .2411 \text{ Wind Speed}.$$

Although the R^2 from this fit is high, $R^2 = .875$, I am unhappy with the fit of the model. The plot of the residuals against the fitted values clearly points out the inadequacy:



The `rvfplot` shows that the linear regression systematically underestimates the DC output for wind speeds in the middle, and overestimates the DC output for low and high wind speeds. This model is not acceptable for making predictions - one can and should do better!

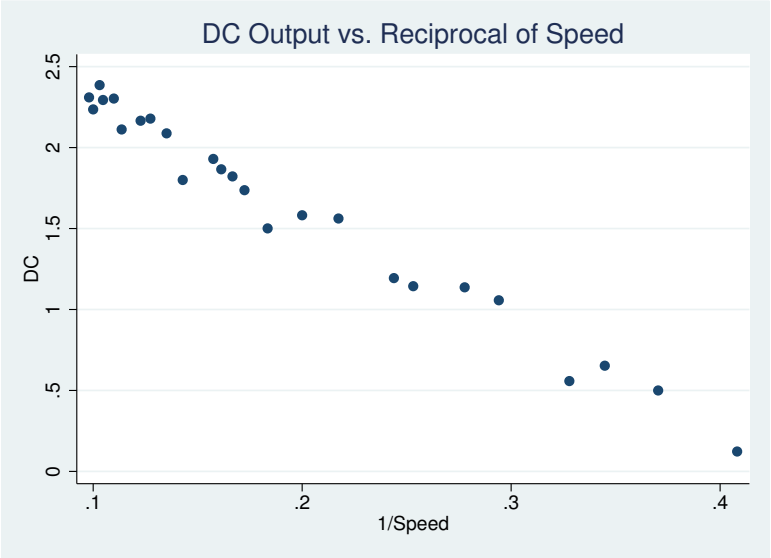
The original data plot indicates that DC output approaches an upper limit of about 2.5 amps as the wind speed increases. Given this fact, and the trend in the plot, I decided to use the inverse of wind speed as a predictor of DC output. Another reasonable first step would be a logarithmic transformation of wind speed but this function steadily increases without approaching a finite limit.

Aside: The above plot is not the same as in the previous notes or in the lab. I decided to illustrate further the flexibility of Stata and the power of `do files`. We obtained exactly those four plots in Minitab if we requested the 4-in-1 plots in regression. You might want to replace the histogram with a boxplot – the modification is simple. The `do file` statements to produce the plot after running the regression command are:

```
predict residual, r
quietly qnorm residual, saving(probplot, replace) nodraw ///
    title(Normal Prob. Plot of Residuals)
quietly rvfplot, saving(respredplot, replace) nodraw ///
    title(Residuals vs. Fitted Values)
quietly hist residual, freq saving(hist, replace) nodraw ///
    title(Histogram of the Residuals)
generate obs_order = _n
quietly twoway connect residual obs_order, saving(obs_order, replace) ///
    nodraw title(Residuals vs. Order of the Data)
drop obs_order
graph combine probplot.gph respredplot.gph hist.gph obs_order.gph, ///
      title(Residual Plots)
```

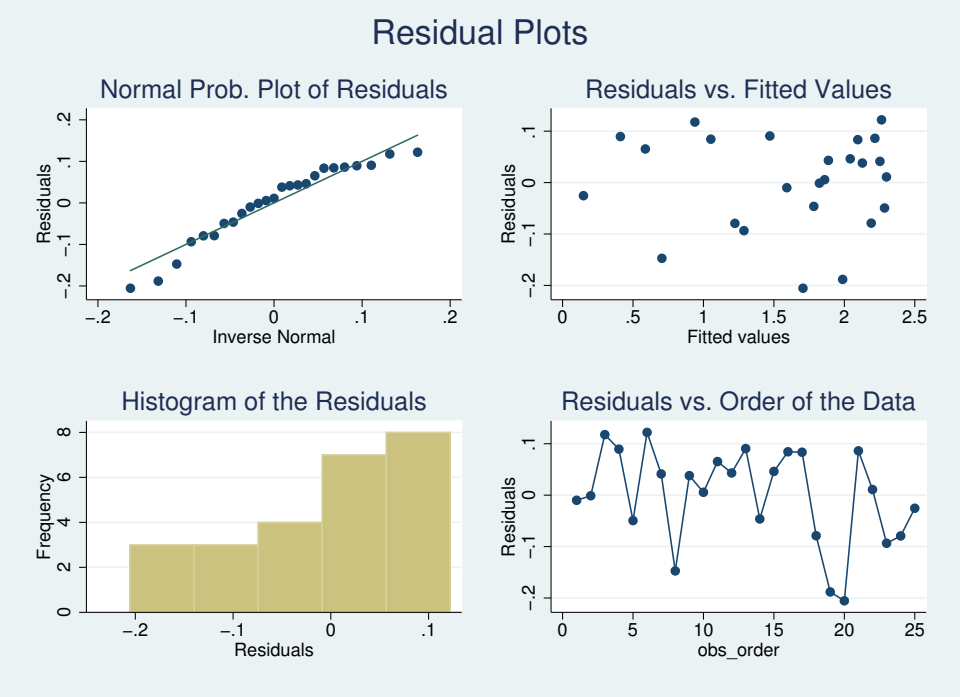
This program will fail if the variable `residual` exists before you run it (that can be fixed).

A plot of DC output against one over the wind speed is fairly linear:



This suggests that a simple linear regression fit on this scale is appropriate. Note that DC output is a decreasing function of one over the wind speed.

```
. regress dc speed_inv
      Source |       SS       df       MS              Number of obs =      25
-------------+------------------------------           F(  1,    23) = 1128.43
       Model |  10.0072178     1  10.0072178           Prob > F      =  0.0000
    Residual |  .203969527    23   .00886824           R-squared     =  0.9800
-------------+------------------------------           Adj R-squared =  0.9792
       Total |  10.2111874    24   .42546614           Root MSE      =  .09417
------------------------------------------------------------------------------
          dc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   speed_inv |  -6.934547   .2064335   -33.59   0.000    -7.361588   -6.507507
       _cons |    2.97886   .0449023    66.34   0.000     2.885973    3.071748
------------------------------------------------------------------------------
```

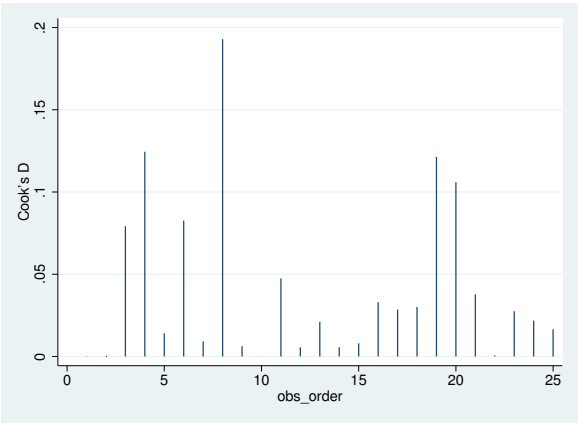


The LS regression line is

$$\text{Predicted DC output} = 2.9789 - 6.9345\frac{1}{\text{Wind speed}}.$$

The residual plots show left skewness, but no serious outliers. The Shapiro-Wilk test has a p-value of 0.08. The transformation appears to work well, although if I tried harder I might be able to symmetrize the residuals a little better (I would start by transforming Y instead of X). I don't think it is worth the trouble here, though. It is fairly clear by examining the scatter plot (the one corresponding to the actual regression we did!) that there are no highly influential points here. Still, we really should check the Cook's D values as a routine matter. Since 1 is a common cutoff for Cook's D, and no values stand out much, we have little to be concerned over.

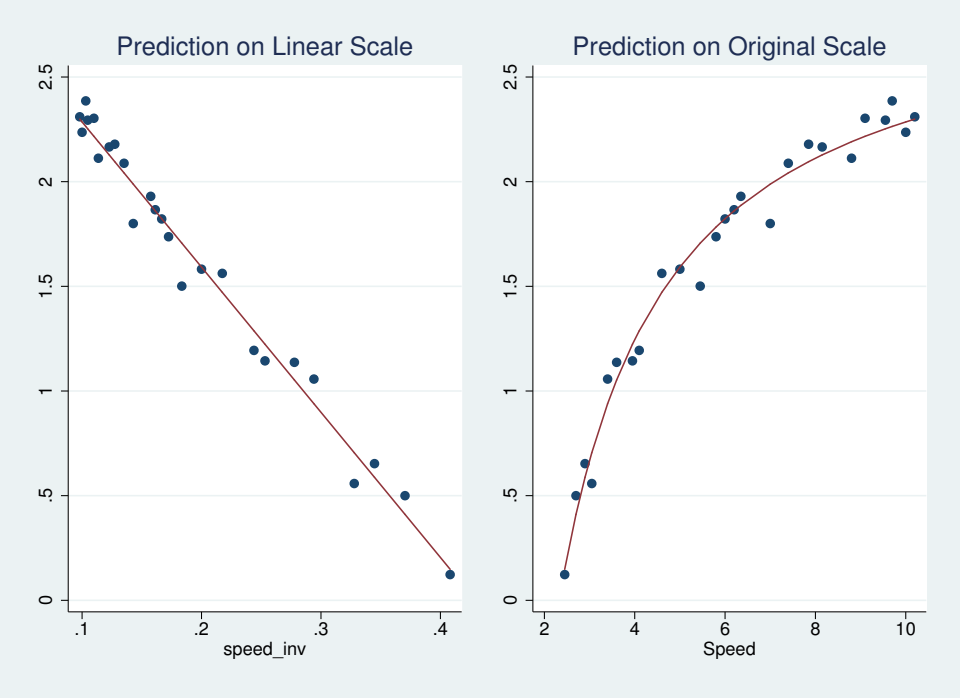
```
. predict cooksd,cooksd
. gene obs_order = _n
. twoway spike cooksd obs_order
```



All our theory and modelling applies in the linear scale (the transformed problem where we fit output to 1/speed). We really want to see how well things appear to work in the original scale, though. The following statements accomplish that.

```
. regress dc speed_inv
. predict pred_dc,xb
. twoway (scatter dc speed_inv) (line pred_dc speed_inv,sort),legend(off)
> title(Prediction on Linear Scale) saving(l,replace)
. twoway (scatter dc speed) (line pred_dc speed,sort),legend(off)
> title(Prediction on Original Scale) saving(o,replace)
. graph combine l.gph o.gph
```

We would put confidence and prediction bands on the plot in a similar manner. How would we predict output (with a prediction interval) for a wind speed of 15?



Brain Weights and Body Weights of Mammals

The data below are the average brain weight (g) and body weights (kg) for 62 species of mammals. We are interested in developing a model for predicting brain weight from body weight.

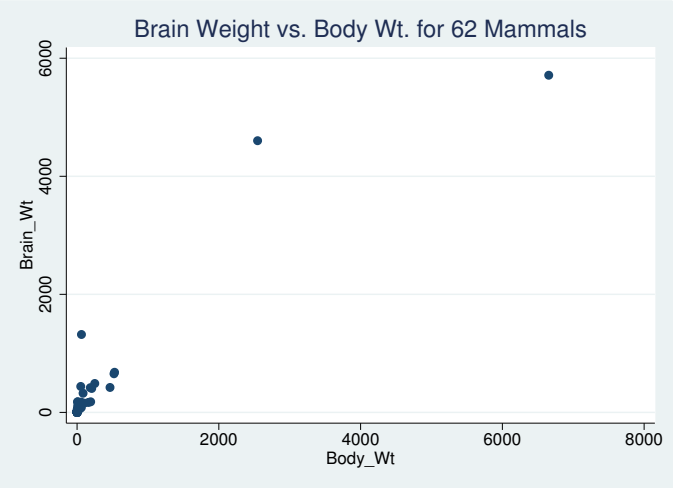
```
. list,clean
```

	species	body_wt	brain_wt
1.	Arctic fox	3.385	44.5
2.	Owl monkey	.48	15.499
3.	Mountain beaver	1.35	8.1
4.	Cow	465	423
5.	Gray wolf	36.33	119.5
6.	Goat	27.66	115
7.	Roe deer	14.83	98.2
8.	Guinea pig	1.04	5.5
9.	Vervet	4.19	58
10.	Chinchilla	.425	6.4
11.	Ground squirrel	.101	4
12.	Arctic ground squirrel	.92	5.7
13.	Africa giant poached rat	1	6.6
14.	Lesser short-tailed shrew	.005	.14
15.	Star-nosed mole	.06	1
16.	Nine-banded armadillo	3.5	10.8
17.	Tree hyrax	2	12.3
18.	N. American opussum	1.7	6.3
19.	Asian elephant	2547	4603
20.	Big brown bat	.023	.3
21.	Donkey	187.1	419
22.	Horse	521	655
23.	European hedgehog	.785	3.5
24.	Patas monkey	10	115
25.	Cat	3.3	25.6
26.	Galago	.2	5
27.	Genet	1.41	17.5
28.	Giraffe	529	680
29.	Gorilla	207	406
30.	Gray seal	85	325
31.	Rock hyrax	.75	12.3
32.	Human	62	1320
33.	African elephant	6654	5712
34.	Water opussum	3.5	3.9
35.	Rhesus monkey	6.8	179
36.	Kangaroo	35	56
37.	Yellow-bellied marmot	4.05	17
38.	Golden hamster	.12	1
39.	Mouse	.023	.4
40.	Little brown bat	.01	.25
41.	Slow loris	1.4	12.5
42.	Okapi	250.01	490
43.	Rabbit	2.5	12.1
44.	Sheep	55.5	175
45.	Jaguar	100	157
46.	Chimpanzee	52.16	440
47.	Baboon	10.55	179.5
48.	Desert hedgehog	.55	2.4
49.	Giant armadillo	60	81
50.	Rock hyrax	3.6	21
51.	Raccoon	4.288	39.2
52.	Rat	.28	1.9
53.	Eastern American mole	.075	1.2
54.	Mole rat	.122	3
55.	Musk shrew	.048	.33
56.	Pig	192	180
57.	Echidna	3	25
58.	Brazilian tapir	160	169
59.	Tenrec	.9	2.6
60.	Phalanger	1.62	11.4

```
61.                 Tree shrew      .104        2.5
62.                    Red fox    4.235       50.4
```

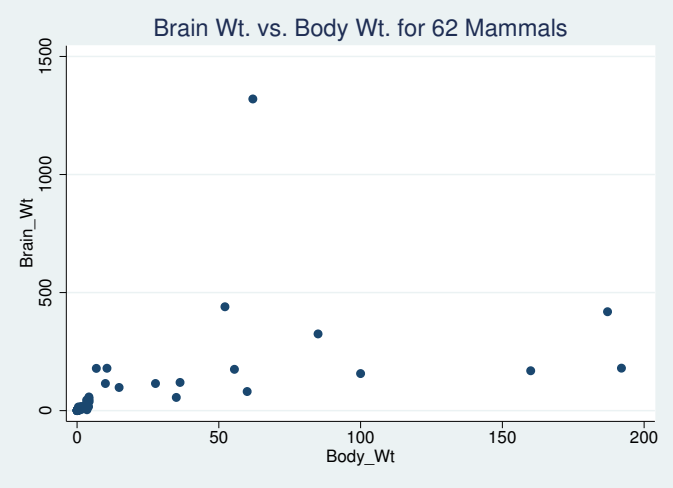
A plot of the brain weights against the body weights is non-informative because many species have very small brain weights and body weights compared to the elephants:

```
. scatter br bo,tit(Brain Weight vs. Body Wt. for 62 Mammals)
```



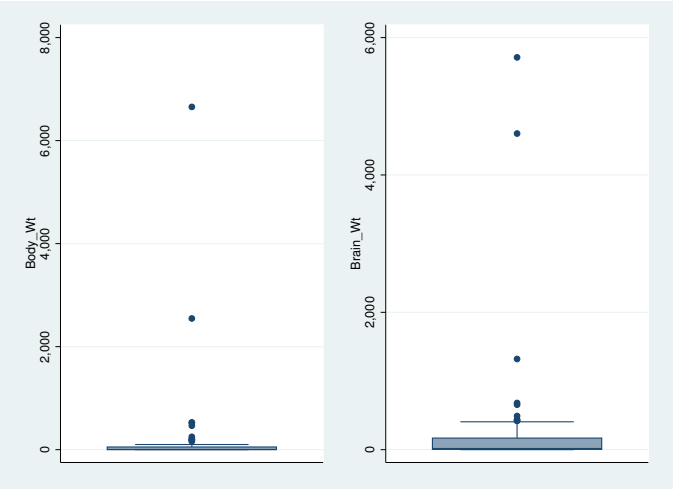
If we momentarily hold out the species with body weights exceeding $200kg$ or brain weights exceeding $200g$, and replot the data, we see that the brain weight of mammals typically increases with the body weight, but the relationship is nonlinear:

```
. scatter br bo if(bo<=200),tit(Brain Wt vs. Body Wt. for 62 Mammals)
```



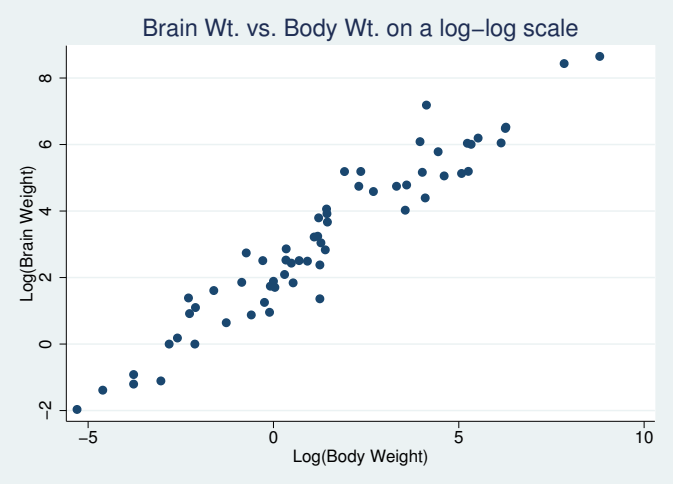
The trend suggests transforming both variables to a logarithmic scale to linearize the relationship between brain weight and body weight. It does not matter which base logarithm you choose. The relationship is no more linear with one base than another. I will use natural logarithms. What is even more compelling about the log transform here is the extreme *right* skewness of both variables – logs pull extremely large values down much more than more modest values, so they tend to symmetrize such data (and regression works much better when both variables have reasonably symmetric distributions).

```
. graph box bod,name(bodbox)
. graph box br,name(brbox)
. graph combine bodbox brbox
```



The plot of $\log_e$(brain weight) against $\log_e$(body weight) is fairly linear:

```
. gene lbod=log(body_wt)
. gene lbr = log(brain_wt)
. scatter lbr lbod,title(Brain Wt. vs. Body Wt. on a log-log scale) xti(Log(Bod
> y Weight)) yti(Log(Brain Weight))
```



At this point I considered fitting the model:

$$\log_e(\text{brain weight}) = \beta_0 + \beta_1 \log_e(\text{body weight}) + \epsilon.$$

Summary information from fitting this model:

```
. regre lbr lbo

      Source |       SS       df       MS              Number of obs =      62
-------------+------------------------------           F(  1,    60) =  697.42
       Model |  336.188164     1  336.188164            Prob > F      =  0.0000
    Residual |  28.9225677    60  .482042795            R-squared     =  0.9208
-------------+------------------------------           Adj R-squared =  0.9195
       Total |  365.110732    61  5.98542184            Root MSE      =  .69429

------------------------------------------------------------------------------
         lbr |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        lbod |   .7516859   .0284635    26.41   0.000     .6947505    .8086214
       _cons |   2.134787   .0960432    22.23   0.000     1.942672    2.326902
------------------------------------------------------------------------------
```

The fitted relationship:

$$\text{Predicted } \log_e(\text{brain weight}) = 2.135 + 0.752 \log_e(\text{body weight}),$$

explains about 92% of the variation in $\log_e$(brain weight). The $t-$test for $H_0 : \beta_1 = 0$ is highly significant ($p - value = 0$ to three decimal places). This summary information combined with the data plot indicates that there is a strong linear relationship between $\log_e$(brain weight) and $\log_e$(body weight), with the average $\log_e$(brain weight) increasing as $\log_e$(body weight) increases.

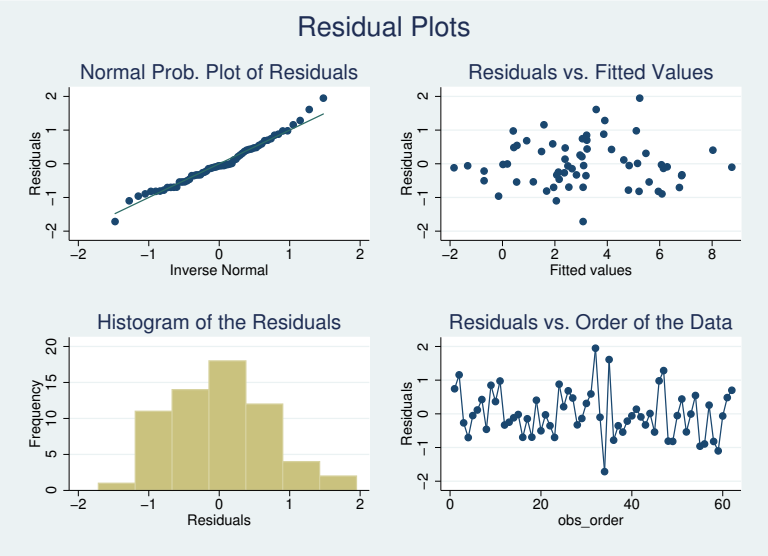
To predict brain weights, use the inverse transformation

$$\text{Predicted brain weight} = \exp\{\text{Predicted } \log_e(\text{brain weight})\}$$

or

$$\begin{aligned} \text{Predicted brain weight} &= \exp\{2.135 + 0.752 \log_e(\text{body weight})\} \\ &= \exp(2.135) * \text{body weight}^{0.752} \\ &= 8.457 * \text{body weight}^{0.752}. \end{aligned}$$

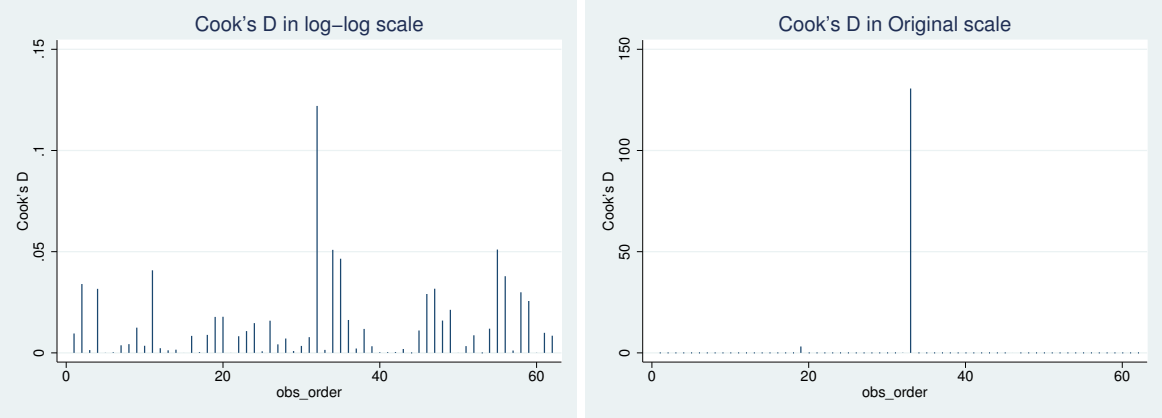
These conclusions are tentative, subject to a careful residual analysis. Residual plots do not suggest any serious deficiencies with the model, but do highlight one or more poorly fitted species:



Can anyone guess what species these may be, and what further analyses might be reasonable? The largest and smallest residuals belong to observations 32 and 34 respectively (obtained from simply entering the data editor). Note that a normal probability (or $Q-Q$) plot of the residuals is reasonably straight and the Shapiro-Wilk test of normality indicates no gross departures from normality:

```
                    Shapiro-Wilk W test for normal data
    Variable |    Obs        W          V          z     Prob>z
-------------+-------------------------------------------------
         res |     62    0.98268      0.967     -0.073  0.52927
```

Cook's D does not show any particular problems (until the value approaches 1, most data analysts do not worry much about it). Compare it to the value in the original scale where the distribution of both variables was so skewed.



Usually it is worth plotting the fitted values back on the original scale as we did for the wind speed data. That would not be very useful here since the original scale obscures most of the data.