Appendix A: Estimating a Power-Law Regression in Stata

Recall the power-law function:

$$y = a \cdot x^k$$

We need to estimate a and k for each group and then statistically compare them. But first consider estimating them for a single group, where y is a variable in our Stata dataset titled "Performance", x is a variable in the dataset titled "Trial", and "ParticipantID" is a variable indicating the participant. We then use Stata's nonlinear regression command (nl) as follows:

```
nl (Performance = {a0}*Trial^{k0}), vce(cluster ParticipantID)
```

Below is an example regression with the associated output.

```
. nl (Performance = {a}*Trial^{k}) if IV == 0, vce(cluster Person)

Iteration 0:  residual SS =  5480.425
Iteration 1:  residual SS =  3127.483
Iteration 2:  residual SS =  2141.798
Iteration 3:  residual SS =  1823.836
Iteration 4:  residual SS =   1823.82
Iteration 5:  residual SS =   1823.82


Nonlinear regression                              Number of obs =      1,800
                                                  R-squared     =     0.9885
                                                  Adj R-squared =     0.9882
                                                  Root MSE      =   1.007155
                                                  Res. dev.     =   5131.843

                                   (Std. err. adjusted for 36 clusters in Person)
```

Performance	Coefficient	Robust std. err.	t	P>|t|	[95% conf.	interval]
/a	5.084096	.0489309	103.90	0.000	4.984761	5.183431
/k	.1960677	.0028545	68.69	0.000	.1902727	.2018626

Moving to multiple groups is now easy. Assume that we have two groups, indicated by the "Group" variable, which takes on a value of 0 for one group and 1 for the other group. We can estimate the following regression:

```
nl (Performance = ({a0}+{a1}*Group)*Trial^({k0}+{k1}*Group), vce(cluster ParticipantID)
```

```
. nl (Performance = ({a0}+{a1}*IV)*Trial^({k0}+{k1}*IV)), vce(cluster Person)

Iteration 0:  residual SS =  129008.9
Iteration 1:  residual SS =  107326.8
Iteration 2:  residual SS =  57447.98
Iteration 3:  residual SS =  4499.407
Iteration 4:  residual SS =   3713.24
Iteration 5:  residual SS =  3687.216
Iteration 6:  residual SS =  3687.216
Iteration 7:  residual SS =  3687.216

Nonlinear regression                              Number of obs =      3,650
                                                  R-squared     =     0.9972
                                                  Adj R-squared =     0.9972
                                                  Root MSE      =   1.005636
                                                  Res. dev.     =   10395.28

                                (Std. err. adjusted for 73 clusters in Person)
```

Performance	Coefficient	Robust std. err.	t	P>|t|	[95% conf.	interval]
/a0	5.084096	.0485868	104.64	0.000	4.98724	5.180952
/a1	-.0777175	.05949	-1.31	0.196	-.1963087	.0408737
/k0	.1960677	.0028344	69.17	0.000	.1904173	.201718
/k1	.3032024	.0034596	87.64	0.000	.2963059	.310099

```
Y-intercept when IV == 1

. nlcom (_b[a0:_cons]+_b[a1:_cons])

       _nl_1:  _b[a0:_cons]+_b[a1:_cons]
```

Performance	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_nl_1	5.006379	.0343276	145.84	0.000	4.939098	5.07366

```
Growth rate when IV == 1

. nlcom (_b[k0:_cons]+_b[k1:_cons])

       _nl_1:  _b[k0:_cons]+_b[k1:_cons]
```

Performance	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_nl_1	.4992701	.0019836	251.70	0.000	.4953823	.5031579

nl (Performance = ({a0}+{a1}*IV1+{a2}*IV2+{a3}*IV1_IV2)*Trial({k0}+{k1}*IV1+{k2}*IV2+{k3}*IV1_IV2), vce(cluster ParticipantID)

a3 and k3 are the interaction terms for the y-intercept and growth rate, respectively. The y-intercept and growth rate for each condition can be calculated as follows (using nlcom):

- [Y-intercept | IV1 == 0 & IV2 == 0] = a0
- [Y-intercept | IV1 == 1 & IV2 == 0] = a0 + a1
- [Y-intercept | IV1 == 0 & IV2 == 1] = a0 + a2

- [Y-intercept | IV1 == 1 & IV2 == 1] = a0 + a1 + a2 + a3
- [Y-intercept | IV1 == 0 & IV2 == 0] = k0
- [Y-intercept | IV1 == 1 & IV2 == 0] = k0 + k1
- [Y-intercept | IV1 == 0 & IV2 == 1] = k0 + k2
- [Y-intercept | IV1 == 1 & IV2 == 1] = k0 + k1 + k2 + k3

Log Transformation

Rather than estimating the power-law function directly, some papers log both sides of the equation and run a regular OLS regression (Levitt et al. 2013). That is, they take the natural log of performance and the trial variable and then simply use the reg command. Note the additive form of the log of the power-law function:

$$\ln(y) = \ln(a \cdot x^k) \rightarrow \ln(y) = \ln(a) + k * \ln(x)$$

Thus, ln(a) is the y-intercept and k is the linear increase on the log-transformed scale. This seems to be an elegant method. However, taking a natural log can be problematic because it is sensitive to additive transformations of the dependent or independent variables, which are necessary when there are values of 0 or less. And I don't see any reason to use the log approach given the availability of nlreg.