SSCC social science computing cooperative

Research Computing

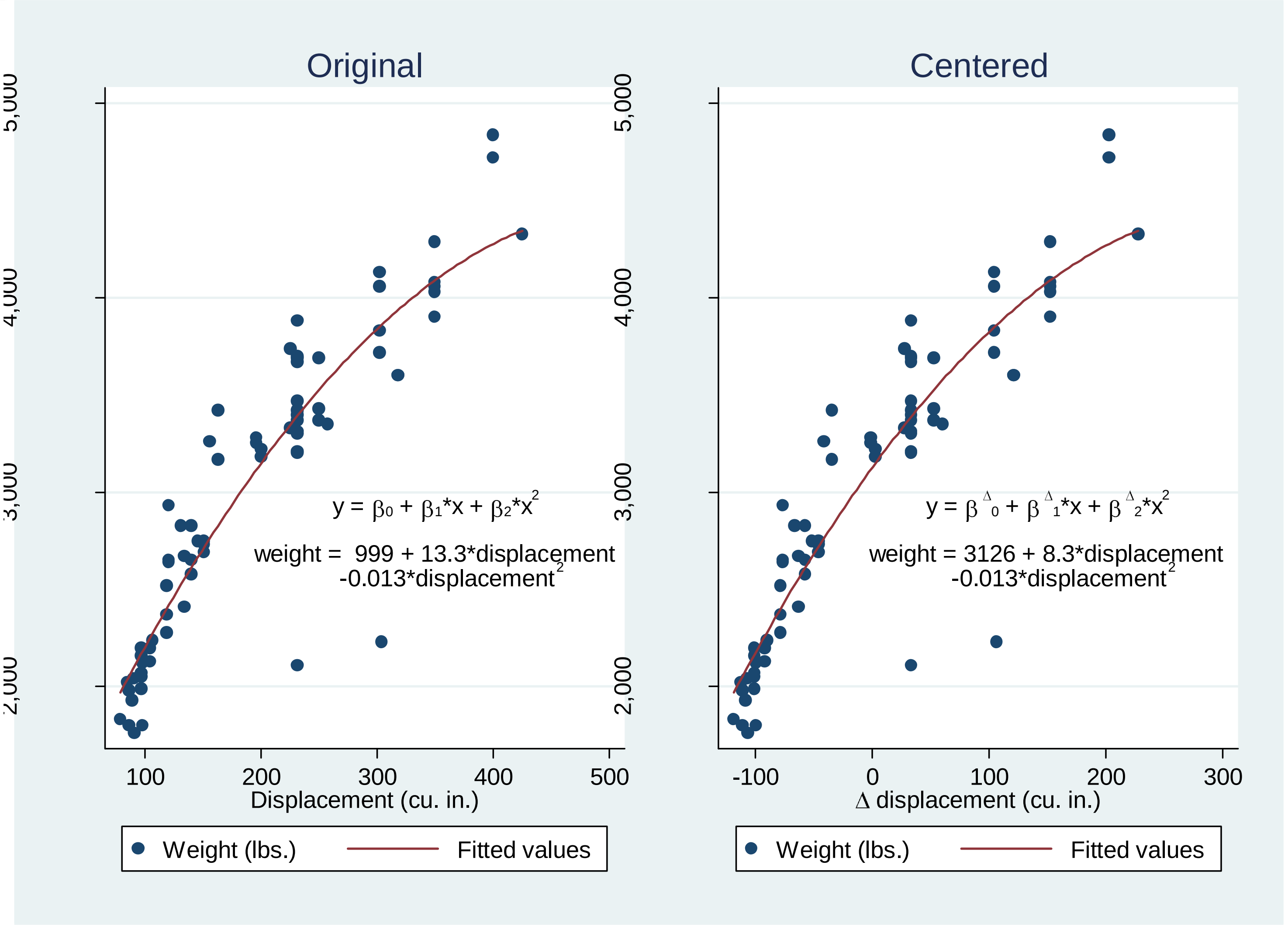
University of Wisconsin – Madison

Post-estimation Parameter Recentering and Rescaling

Doug Hemken

SSCC social science computing cooperative

Recentered polynomial regression (change of basis)



Recentered polynomial regression (change of basis)

```
. estimates table Original Centered, se

----------------------------------------------
     Variable |  Original      Centered
--------------+-------------------------------
 displacement |  13.292618     8.2613721
              |  2.1114091     .49321693
              |
c.displacement#|
c.displacement | -.01275042   -.01275042
              |  .00461032     .00461032
              |
        _cons |  999.27223     3125.5442
              |  211.52293     54.591876
----------------------------------------------
                                legend: b/se
```

Math – Linear Algebra of Recentering and Rescaling

- Building Blocks – Simple regression models
  - Recentering
  - Rescaling
- Adding Interactions – Factorial regression models
  - Full factorial
  - Partial
- Adding Categorical terms
  - Untransformed
  - Recentering via contrasts
- Group like terms – Polynomial models

Simple regression recentering

- Given a model
  - $y = \beta_0 + \beta_1 x$
- And a recentering constant
  - $\Delta x = x - \boldsymbol{\mu}$
- Then the recentered model
  - $y = \beta_0^{\Delta} + \beta_1^{\Delta} \Delta x$
- Has parameters given by
  - $B^{\Delta} = \begin{bmatrix} 1 & \boldsymbol{\mu} \\ 0 & 1 \end{bmatrix} B$, or
  - $\begin{bmatrix} \beta_0^{\Delta} \\ \beta_1^{\Delta} \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{\mu} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$

Precision matrices

- Let the parameter transformation be given by
  - $C = \begin{bmatrix} 1 & \mu \\ 0 & 1 \end{bmatrix}$

- Given the precision matrix for the original model, V, then the precision matrix of the recentered model is
  - $V_\Delta = CVC'$

Recentering y

- Given
  - $y = \beta_0 + \beta_1 x$
  - $\Delta x = x - \mu_x$
  - $\Delta y = y \boldsymbol{- \mu_y}$
- Then
  - $\Delta y = \beta_0^{\Delta \mathrm{y}} + \beta_1^{\Delta} \Delta x$
- Is
  - $B^{\Delta \mathrm{y}} = \begin{bmatrix} 1 & \mu_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \boldsymbol{+ \mu_y} \\ \beta_1 \end{bmatrix}$

Simple regression rescaling

- Given a model
  - $y = \beta_0 + \beta_1 x$
- And a rescaling constant
  - $z = x/\boldsymbol{\sigma}$
- Then the rescaled model
  - $y = \beta_0^z + \beta_1^z z$
- Has parameters given by
  - $B^z = \begin{bmatrix} 1 & 0 \\ 0 & \boldsymbol{\sigma} \end{bmatrix} B$

Rescaling y

- From
  - $y = \beta_0 + \beta_1 x$
  - $z = x/\sigma_x$
  - $y_s = y/\boldsymbol{\sigma_y}$
- To
  - $y_s = \beta_0^{zy} + \beta_1^{z} z$
- Is
  - $B^{zy} = \frac{1}{\boldsymbol{\sigma_y}} \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix} B$

Standardizing x

- Combine the two simpler transformations

$$B^{std} = \begin{bmatrix} 1 & 0 \\ 0 & \boldsymbol{\sigma} \end{bmatrix} \begin{bmatrix} 1 & \boldsymbol{\mu} \\ 0 & 1 \end{bmatrix} B$$

Factorial model recentering

- Given
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2$
  - $\Delta x_1 = x_1 \; \boldsymbol{-\mu_1}$ and $\Delta x_2 = x_2 \; \boldsymbol{-\mu_2}$
- Then
  - $y = \beta_0^\Delta + \beta_1^\Delta \Delta x_1 + \beta_2^\Delta \Delta x_2 + \beta_{12}^\Delta \Delta x_1 \Delta x_2$
  - (variable-wise centered, not term-wise centered)
- Is given by

$$B^\Delta = \begin{bmatrix} 1 & \boldsymbol{\mu_2} \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & \boldsymbol{\mu_1} \\ 0 & 1 \end{bmatrix} B = \begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_1\mu_2 \\ 0 & 1 & 0 & \mu_2 \\ 0 & 0 & 1 & \mu_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{12} \end{bmatrix}$$

Kronecker ("direct") products

- Let

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \text{ and } B = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix}$$

- Then

$$A \otimes B = \begin{bmatrix} a_1 B & a_2 B \\ a_3 B & a_4 B \end{bmatrix}$$

$$= \left[\begin{array}{cc|cc} a_1 b_1 & a_1 b_2 & a_2 b_1 & a_2 b_2 \\ a_1 b_3 & a_1 b_4 & a_2 b_3 & a_2 b_4 \\ \hline a_3 b_1 & a_3 b_2 & a_4 b_1 & a_4 b_2 \\ a_3 b_3 & a_3 b_4 & a_4 b_3 & a_4 b_4 \end{array}\right]$$

Factorial model rescaling

- Given
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2$
  - $z_1 = x_1/\boldsymbol{\sigma_1}$ and $z_2 = x_2/\boldsymbol{\sigma_2}$

- Then
  - $y = \beta_0^z + \beta_1^z z_1 + \beta_2^z z_2 + \beta_{12}^z z_1 z_2$

- Is given by
  - $B^z = \begin{bmatrix} 1 & 0 \\ 0 & \boldsymbol{\sigma_2} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & \boldsymbol{\sigma_1} \end{bmatrix} B$
  - $B^z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \boldsymbol{\sigma_1} & 0 & 0 \\ 0 & 0 & \boldsymbol{\sigma_2} & 0 \\ 0 & 0 & 0 & \boldsymbol{\sigma_1 \sigma_2} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{12} \end{bmatrix}$

Three-way recentering

- Given
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_3 x_3 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \beta_{123} x_1 x_2 x_3$
  - $\Delta x_1 = x_1 - \boldsymbol{\mu_1}, \Delta x_2 = x_2 - \boldsymbol{\mu_2}, \Delta x_3 = x_3 - \boldsymbol{\mu_3}$
- Then

$$B^{\Delta} = \begin{bmatrix} 1 & \mu_3 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & \mu_2 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & \mu_1 \\ 0 & 1 \end{bmatrix} B$$

$$= \begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_1\mu_2 & \mu_3 & \mu_1\mu_3 & \mu_2\mu_3 & \mu_1\mu_2\mu_3 \\ 0 & 1 & 0 & \mu_2 & 0 & \mu_3 & 0 & \mu_2\mu_3 \\ 0 & 0 & 1 & \mu_1 & 0 & 0 & \mu_3 & \mu_1\mu_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \mu_3 \\ 0 & 0 & 0 & 0 & 1 & \mu_1 & \mu_2 & \mu_1\mu_2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \mu_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mu_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{12} \\ \beta_3 \\ \beta_{13} \\ \beta_{23} \\ \beta_{123} \end{bmatrix}$$

Partial Factorial

- Suppose a model has only 2nd order interaction terms
- This is $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_3 x_3 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \beta_{123} x_1 x_2 x_3$ with $\boldsymbol{\beta_{123} = 0}$. In our centered model, likewise, we have $\boldsymbol{\beta_{123}^{\Delta} = 0}$
- Then we can simplify our notation:

$$\begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_1\mu_2 & \mu_3 & \mu_1\mu_3 & \mu_2\mu_3 & \mu_1\mu_2\mu_3 \\ 0 & 1 & 0 & \mu_2 & 0 & \mu_3 & 0 & \mu_2\mu_3 \\ 0 & 0 & 1 & \mu_1 & 0 & 0 & \mu_3 & \mu_1\mu_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \mu_3 \\ 0 & 0 & 0 & 0 & 1 & \mu_1 & \mu_2 & \mu_1\mu_2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \mu_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mu_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{12} \\ \beta_3 \\ \beta_{13} \\ \beta_{23} \\ \boldsymbol{\beta_{123}} \end{bmatrix}$$

$$\rightarrow simplifies\ as \begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_1\mu_2 & \mu_3 & \mu_1\mu_3 & \mu_2\mu_3 \\ 0 & 1 & 0 & \mu_2 & 0 & \mu_3 & 0 \\ 0 & 0 & 1 & \mu_1 & 0 & 0 & \mu_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \mu_1 & \mu_2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{12} \\ \beta_3 \\ \beta_{13} \\ \beta_{23} \end{bmatrix}$$

Additive models again

- Suppose a model has only 1st order terms, like
  - $y = \beta_0 + \beta_1 x_1 + \beta_3 x_3$
- This is
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_3 x_3 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \beta_{123} x_1 x_2 x_3$, with $many\ zeros$.
- Then we can vastly simplify our notation:

$$\begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_1\mu_2 & \mu_3 & \mu_1\mu_3 & \mu_2\mu_3 & \mu_1\mu_2\mu_3 \\ 0 & 1 & 0 & \mu_2 & 0 & \mu_3 & 0 & \mu_2\mu_3 \\ 0 & 0 & 1 & \mu_1 & 0 & 0 & \mu_3 & \mu_1\mu_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \mu_3 \\ 0 & 0 & 0 & 0 & 1 & \mu_1 & \mu_2 & \mu_1\mu_2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \mu_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mu_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{12} \\ \beta_3 \\ \beta_{13} \\ \beta_{23} \\ \beta_{123} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

Factor variables

- Suppose g is a factor with three categories, and x_1 and x_2 are as before
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{g_1} g_1 + \beta_{1g_1} g_1 x_1 + \beta_{2g_1} g_1 x_2 + \beta_{12g_1} g_1 x_1 x_2 + \beta_{g_2} g_2 + \beta_{1g_2} g_2 x_1 + \beta_{2g_2} g_2 x_2 + \beta_{12g_2} g_2 x_1 x_2$
  - `.regress y i.g##c.x1##c.x2`

  - With reference coding (this is also a direct sum),
  - $B^{\Delta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & \mu_2 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & \mu_1 \\ 0 & 1 \end{bmatrix} B$

Block diagonal, or direct sum

$$\left[\begin{array}{cccc|cccc|cccc}
1 & \mu_1 & \mu_2 & \mu_1\mu_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & \mu_1 & \mu_2 & \mu_1\mu_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & \mu_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & \mu_1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mu_1 & \mu_2 & \mu_1\mu_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \mu_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mu_1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

Factor Grand Mean Centering

- To transform from reference coding to grand mean centered coding, the transformation matrix depends on the number of categories:
- Two categories are centered by

$$\begin{bmatrix} 1 & 1/2 \\ 0 & -1/2 \end{bmatrix}$$

- Three categories

$$\begin{bmatrix} 1 & 1/3 & 1/3 \\ 0 & 2/3 & -1/3 \\ 0 & -1/3 & 2/3 \end{bmatrix}$$

- Four categories

$$\begin{bmatrix} 1 & 1/4 & 1/4 & 1/4 \\ 0 & 3/4 & -1/4 & -1/4 \\ 0 & -1/4 & 3/4 & -1/4 \\ 0 & -1/4 & -1/4 & 3/4 \end{bmatrix}$$

Grand Mean transformation

- For n categories:

$$\begin{bmatrix} 1 & 1/n & \cdots & 1/n \\ 0 & \dfrac{n-1}{n} & -1/n & -1/n \\ \vdots & -1/n & \ddots & \vdots \\ 0 & -1/n & \cdots & \dfrac{n-1}{n} \end{bmatrix}$$

Polynomial terms

- Now consider a model of the form
  - $y = \beta_0 + \beta_1 x + \beta_{12} x^2$
- Which we will rewrite as
  - $y = \beta_0 + \beta_1 x + \beta_{12} xx$
- In Stata we could specify such a model as
  - `regress y c.x##c.x`

Polynomial Terms

- Here we'll need to collect terms

  If $A = \begin{bmatrix} 1 & \mu_1 \\ 0 & 1 \end{bmatrix}$ then

$$A \otimes A = \begin{bmatrix} 1 & \mu_1 & \mu_1 & \mu_1\mu_1 \\ 0 & 1 & 0 & \mu_1 \\ 0 & 0 & 1 & \mu_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- However, this is a matrix that starts with two β_1 and returns two β_1^{Δ}.

$$\begin{bmatrix} \beta_0^{\Delta} \\ \beta_1^{\Delta} \\ \beta_1^{\Delta} \\ \beta_{12}^{\Delta} \end{bmatrix} = \begin{bmatrix} 1 & \mu_1 & \mu_1 & \mu_1\mu_1 \\ 0 & 1 & 0 & \mu_1 \\ 0 & 0 & 1 & \mu_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_1 \\ \beta_{12} \end{bmatrix}$$

Polynomial Terms

- Letting one $\beta_1 = 0$, we simplify our matrix to

$$\begin{bmatrix} \beta_0^\Delta \\ \beta_1^\Delta \\ \beta_1^\Delta \\ \beta_{12}^\Delta \end{bmatrix} = \begin{bmatrix} 1 & \mu_1 & \mu_1\mu_1 \\ 0 & 1 & \mu_1 \\ 0 & 0 & \mu_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_{12} \end{bmatrix}$$

- But from here, we need to collect our β_1^Δ terms

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \mu_1 & \mu_1\mu_1 \\ 0 & 1 & \mu_1 \\ 0 & 0 & \mu_1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \mu_1 & \mu_1^2 \\ 0 & 1 & 2\mu_1 \\ 0 & 0 & 1 \end{bmatrix}$$

So

$$B^\Delta = \begin{bmatrix} 1 & \mu_1 & \mu_1^2 \\ 0 & 1 & 2\mu_1 \\ 0 & 0 & 1 \end{bmatrix} B$$

Math Summary

- We have building blocks for:
  - Continuous variables
  - Categorical variables
  - Polynomial terms
- We can combine them as:
  - Factorial models
  - Subsets of terms from factorial models
    - *(As long as no higher-order terms appear without their related lower-order terms)*

Programming – Stata

- Given a model in Stata, we want to
  - Identify variables, variable types, variables' polynomial degree (macro list functions and **`_ms_parse_parts`**)
  - Collect recentering and rescaling constants (**`tabstat`**)
  - Form factorial transformation matrices for continuous/polynomial terms (Kronecker matrix operator, **`#`**)
  - Build complete model transformation matrices by filling constants into the appropriate slots (**`matrix`** extraction and substitution)
  - Use the results (**`estimates store`** and **`estimates table`**)

Kronecker product terms

- In the **`matrix`** language, Kronecker products make it easy to track terms

```
. matrix list A
A[2,2]
                 _      weight
r1               1   3019.4595
r2               0           1

. matrix list B
B[2,2]
                      _  displacement
r1                    1      197.2973
r2                    0             1

. matrix C = B#A
```

Kronecker product terms

- ◦ Column/row names are returned with the form equation(B):name(A)

```
. matrix list C

C[4,4]
                                      displacem~t:  displacem~t:
                   _          weight             _        weight
r1:r1              1       3019.4595      197.2973     595731.19
r1:r2              0               1             0      197.2973
r2:r1              0               0             1     3019.4595
r2:r2              0               0             0             1
```

- Note the **name** stripe is used, but the **equation** stripe is lost.

Combine term parts

- To use this further, we move all the variable names into the name stripe

```
. local cn : colfullnames C
. local cn :subinstr local cn ":" "#", all
. local cn :subinstr local cn "#_" "", all
. matrix coleq C = ""
. matrix colnames C =`cn'
. matrix list C
C[4,4]
                                                  c.displace~t#
                     _        weight  displacement       c.weight
r1:r1                1     3019.4595      197.2973      595731.19
r1:r2                0             1             0       197.2973
r2:r1                0             0             1      3019.4595
r2:r2                0             0             0              1
```

  - Note **`matrix`** understands these are interaction terms!

Kronecker product terms

- And we can keep building …

```
. matrix C = D#C

. matrix list C
```

C[8,8]

	_	weight	displacement	c.displace~t# c.weight	mpg	c.mpg# c.weight	c.mpg# c.displace~t	c.mpg# c.displace~t# c.weight
r1:r1	1	3019.4595	197.2973	595731.19	21.297297	64306.326	4201.8992	12687464
r1:r2	0	1	0	197.2973	0	21.297297	0	4201.8992
r1:r1	0	0	1	3019.4595	0	0	21.297297	64306.326
r1:r2	0	0	0	1	0	0	0	21.297297
r2:r1	0	0	0	0	1	3019.4595	197.2973	595731.19
r2:r2	0	0	0	0	0	1	0	197.2973
r2:r1	0	0	0	0	0	0	1	3019.4595
r2:r2	0	0	0	0	0	0	0	1

Parse covariates from factors

- Use `_ms_parse_parts` with terms from e(b)

```
. quietly regress price foreign##c.weight
. matrix list e(b)

e(b)[1,6]
             0b.          1.                0b.foreign#   1.foreign#
        foreign     foreign       weight    co.weight     c.weight        _cons
y1            0  -2171.5968    2.9948135            0    2.3672266    -3861.719


. _ms_parse_parts weight
. return list // "variable"
scalars:
                 r(omit) =  0
macros:
                 r(name) : "weight"
                 r(type) : "variable"
```

Parse factors from covariates

- Factors

```
. _ms_parse_parts 1.foreign
. return list // "factor"
scalars:
                r(base) =  0
               r(level) =  1
                r(omit) =  0
macros:
                r(name) : "foreign"
                  r(op) : "1"
                r(type) : "factor"
```

Parse interactions

- Interactions

```
. _ms_parse_parts 1.foreign#c.weight
. return list // "interaction"

scalars:
                r(base1) =  0
               r(level1) =  1
              r(k_names) =  2
                 r(omit) =  0
macros:
                r(name2) : "weight"
                  r(op2) : "c"
                r(name1) : "foreign"
                  r(op1) : "1"
                 r(type) : "interaction"
```

Parse polynomials

- Polynomial terms require some extra parsing

```
. _ms_parse_parts whatever#c.whatever

. return list // polynomial as interaction

scalars:
            r(k_names) =  2
               r(omit) =  0

macros:
              r(name2) : "whatever"
                r(op2) : "c"
              r(name1) : "whatever"
                r(op1) : "c"
               r(type) : "interaction"
```

Matrix extraction/substitution

- Recognizes factor notation equivalences!

```
. quietly regress price c.weight##c.disp
. matrix A = e(b)
. matrix B= A[1,1..2] // by numerical index
. matrix B= A[1,"weight"] // by column/row names
. matrix B= A[1,"c.weight#c.displacement"]
. matrix list B

        c.weight#
    c.displace~t
y1      .0143162

. matrix B= A[1,"c.displacement#c.weight"]
. matrix list B

        c.weight#
    c.displace~t
y1      .0143162
```

stdParm syntax

- `stdParm [ , nodepvar store replace`
  `*estimates_table_options*]`

- Produces centered and standardized parameters
- Optionally exclude the response variable
- Results can be stored
- Results can be reported with any `estimates table` options

stdParm use

```
. quietly regress price c.weight##c.mpg

. stdParm

------------------------------------------------------------
    Variable |   Original        Centered      Standardized
-------------+----------------------------------------------
      weight |    5.0670077      .98475137       .25948245
         mpg |    396.78438     -181.98425      -.35696623
             |
   c.weight#|
       c.mpg |   -.19167955     -.19167955      -.29221218
             |
       _cons |   -5944.8806     -686.28559      -.23267895
------------------------------------------------------------
```

stdParm additional statistics

```
. stdParm, stats(N r2) star

-------------------------------------------------------------------
    Variable |    Original            Centered          Standardized
-------------+-----------------------------------------------------
      weight |    5.0670077***        .98475137          .25948245
         mpg |    396.78438*        -181.98425         -.35696623
             |
   c.weight#|
       c.mpg |   -.19167955**       -.19167955**       -.29221218**
             |
       _cons |   -5944.8806         -686.28559         -.23267895
-------------+-----------------------------------------------------
           N |           74                 74                 74
          r2 |    .35969597          .35969597          .35969597
-------------------------------------------------------------------
                            legend: * p<0.05; ** p<0.01; *** p<0.001
```

stdParm after logit

```
. quietly logit foreign c.price##c.weight

. stdParm

------------------------------------------------------------
    Variable |   Original       Centered      Standardized
-------------+----------------------------------------------
       price |    .00331766      .00113549       3.3491337
      weight |   -.00141654     -.00587217      -4.5638148
             |
    c.price#|
    c.weight |   -7.227e-07     -7.227e-07      -1.6566669
             |
       _cons |   -4.5154515     -1.7920268      -1.7920268
------------------------------------------------------------
```

stdParm, eform

```
. quietly logit foreign c.price##c.weight

. stdParm, eform

------------------------------------------------------------
    Variable |   Original        Centered      Standardized
-------------+----------------------------------------------
       price |    1.0033232       1.0011361       28.478052
      weight |    .99858446       .99414503       .01042222
             |
    c.price#|
    c.weight |    .99999928       .99999928       .19077378
             |
       _cons |    .01093867       .16662211       .16662211
------------------------------------------------------------
```

Download/ install

- `net from www.ssc.wisc.edu/~hemken/Stataworkshops`

- Tinker with the source code, suggest improvements:
  https://github.com/Hemken/stdParm