M*plus* Code for Mediation, Moderation and Moderated Mediation Models (1 to 80)

Stride, C.B., Gardner, S.E., Catley, N., & Thomas, F. (2015)

http://www.figureitout.org.uk

This document contains M*plus* Code for testing different configuration of mediation, moderation and moderated mediation models, including those corresponding in type to the 76 configuration listed and index by Andrew Hayes in the documentation for his **SPSS PROCESS macro.** It is recommended that you read his seminal text (Hayes, A. F. (2013). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach*. New York: Guilford Press.) before undertaking such analyses.

Why M*plus*?

Whilst M*plus* requires extra labour and skills, it provides a completely flexible modelling environment that enables you to test unlimited configurations of moderation and mediation, not just the 76 PROCESS models. Specifically, PROCESS is limited to a single primary IV and DV, continuous mediations, either parallel or serial mediation but not both, and dichotomous or continuous outcomes, where the assumptions of standard multiple regression/logistic regression as appropriate are satisfied. M*plus*, on the other hand, can handle all of these scenarios, and can also handle models with multiple IVs and DVs, mediators in serial or in parallel, and models with non-normal DVs. Also, unlike PROCESS, it can handle mediation and moderation where the data structure is multilevel, and can incorporate latent variables.

How do I use the model codes?

Each of the models have the provided diagrams, model equations, and the M*plus* code. The code includes the requisite **DEFINE:**, **ANALYSIS**:, **MODEL**:, and **OUTPUT**: principal commands, as well as preceding **USEVARIABLES:** subcommand that lists hypothetical variables. To apply these examples to your data, you will need to write the **DATA:** and **VARIABLE:** commands and change the hypothetical variable names (e.g. Y, X, M, Z etc.) to match your variable names.

This code and guidance to mediation and moderation testing is designed for people with some basic previous knowledge of M*plus*. Before trying to use this code you need beginner's M*plus* skills, specifically to know how to read your data into M*plus*, how missing data is coded and treated, how models are estimated, how different outcome distributions are specified, how the BY, ON, WITH, and XWITH statements, and the @ and () symbols work, and how MODEL CONSTRAINT: enables functions of parameters to be tested.

If you require M*plus* training/consultation, we offer both public and in-house courses, and consultancy: see www.figureitout.org.uk.

Known potential running issues:

We have found that, compared to the PROCESS macro, the more complex models can take a considerable amount of time to run (upwards of an hour), especially when bootstrapping is used (we tested the models using a relatively powerful laptop with an 8i processor). M*plus* also struggles to fit models (i.e. you get convergence failures) where measures are on scales with a high variance - where this is the case, rescaling predictors e.g. standardising them, usually solves the problem.

Model Template Selection:

Model templates 1 to 76 below match the equivalently numbered models associated with Andrew Hayes' PROCESS macro, albeit with two adaptations:

1) Where Andrew Hayes' templates specify a model and equation generalised from 1 to multiple mediators, then, for the purposes of providing specific example code in Mplus that matches a diagram, the code and diagrams have been written for a model with 2 mediators in mediator only models (4 and 6) and 1 mediator in moderated mediation models. The code can be edited to include as many mediators as is desired
2) All the models and codes exclude covariates, these can be easily added by specifying them as predictors of the outcome and mediators through adding extra ON statements.

Further Models added:

In recognition of models that cannot be fitted by PROCESS, additional models have been added (model 77 onwards). These include mediation with both serial and parallel mediators, moderated serial mediation, and moderated mediation with both serial and parallel mediators. Models will be continually added and updated.

Acknowledgements:

These codes were developed by Dr Chris Stride, with assistance from Sarah Gardner (programming/checking), Nick Catley (equation expansion/checking) and Ffion Thomas (diagram drawing, based on the original PROCESS diagrams by Andrew Hayes).

Model Index

All models have one primary IV and one DV. For the purposes of calculation of the indirect effects and conditional effects:

- The primary IV (variable X) is assumed to be continuous or dichotomous
- Moderators (variables W, V, Q, Z) are assumed to be continuous. The adaptation to handle observed dichotomous moderators can be found at the end of this documentation.
- Mediators (variable M, or M1, M2 etc.) are assumed to be continuous
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression. Examples of how to handle an observed dichotomous DV are provided in model 1e.

Model number	Arrangement of mediators and moderators	Number of distinct mediators	Number of distinct moderators	Page Number
6	2 or more mediators, in series (example uses 2)	2+	0	61
7	1 or more mediators, in parallel if multiple (example uses 1), 1 moderator of IV-Mediator path only	1+	1	64
8	1 or more mediators, in parallel if multiple (example uses 1), 1 moderator moderating both the IV-Mediator path and direct IV-DV path	1+	1	68
9	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the IV-Mediator path only	1+	2	72
10	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating both the IV-Mediator path and direct IV-DV path	1+	2	77
11	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the IV-Mediator path only, all 2-way and 3-way interactions	1+	2	82
12	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the IV-Mediator path and direct IV-DV path, all 2-way and 3-way interactions	1+	2	87
13	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating the IV-Mediator path, 3-way interaction, 1 also moderating direct IV-DV path	1+	2	93
14	1 or more mediators, in parallel if multiple (example uses 1), 1 moderator of Mediator-DV path only	1+	1	99
15	1 or more mediators, in parallel if multiple (example uses 1), 1 moderator of both Mediator-DV and direct IV-DV path	1+	1	103
16	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the Mediator-DV path only	1+	2	108
17	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating both the Mediator-DV and direct IV-DV path	1+	2	113
18	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the Mediator-DV path only, all 2-way and 3-way	1+	2	119

Model number	Arrangement of mediators and moderators	Number of distinct mediators	Number of distinct moderators	Page Number
	interactions			
19	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the Mediator-DV path and direct IV-DV path, all 2-way and 3-way interactions	1+	2	124
20	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating the Mediator- DV path, 3-way interaction, 1 also moderating direct IV-DV path	1+	2	130
21	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, 1 moderating the IV-Mediator path, 1 moderating the Mediator-DV path	1+	2	136
22	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, 1 moderating the IV-Mediator path and direct IV-DV path, 1 moderating the Mediator-DV path	1+	2	141
23	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path, 1 moderating the Mediator-DV path	1+	3	147
24	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating both the IV-Mediator path and direct IV-DV path, 1 moderating the Mediator-DV path	1+	3	155
25	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path with all 2-way and 3-way interactions, 1 moderating the Mediator-DV path	1+	3	164
26	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating both the IV-Mediator path and direct IV-DV path with all 2-way and 3-way interactions, 1 moderating the Mediator-DV path	1+	3	173
27	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path with all 2-way and 3-way interactions, one of which also moderates the direct IV-DV path, 1 moderating the Mediator-DV path	1+	3	183
28	1 or more mediators, in parallel if multiple (example	1+	2	192

Model number	Arrangement of mediators and moderators	Number of distinct mediators	Number of distinct moderators	Page Number
	uses 1), 2 moderators, 1 moderating the IV-Mediator path, 1 moderating the Mediator-DV path and direct IV-DV path			
29	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, 1 moderating the IV-Mediator path, 1 moderating the Mediator-DV path, both moderating the direct IV-DV path	1+	2	198
30	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path, 1 moderating both the Mediator-DV path and the direct IV-DV path	1+	3	204
31	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating both the IV-Mediator path and the direct IV-DV path, 1 moderating both the Mediator-DV path and the direct IV-DV path	1+	3	212
32	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path with all 2-way and 3-way interactions, 1 moderating both the Mediator-DV path and the direct IV-DV path	1+	3	222
33	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating both the IV-Mediator path and the direct IV-DV path with all 2-way and 3-way interactions, 1 moderating both the Mediator-DV path and the direct IV-DV path	1+	3	231
34	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path with all 2-way and 3-way interactions, one of which also moderates the direct IV-DV path, 1 moderating the Mediator-DV path and the direct IV-DV path	1+	3	242
35	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating the Mediator-DV path	1+	3	251
36	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating both the Mediator-DV path and the IV-DV path	1+	3	259

Model number	Arrangement of mediators and moderators	Number of distinct mediators	Number of distinct moderators	Page Number
37	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating the Mediator-DV path with all 2-way and 3-way interactions	1+	3	268
38	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating both the Mediator-DV path and the IV-DV path, with all 2-way and 3-way interactions	1+	3	277
39	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating the Mediator-DV path with all 2-way and 3-way interactions, 1 of which also moderates the direct IV-DV path	1+	3	287
40	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV-Mediator path and the direct IV-DV path, 2 moderating the Mediator-DV path	1+	3	296
41	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV-Mediator path and the direct IV-DV path, 2 moderating both the Mediator-DV path and the direct IV-DV path	1+	3	304
42	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV-Mediator path and the direct IV-DV path, 2 moderating the Mediator-DV path all 2-way and 3-way interactions	1+	3	314
43	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV-Mediator path and the direct IV-DV path, 2 moderating both the Mediator-DV path and the direct IV-DV path with all 2-way and 3-way interactions	1+	3	323
44	1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV-Mediator path and the direct IV-DV path, 2 moderating the Mediator-DV path with all 2-way and 3-way interactions, 1 of which also moderates the direct IV-DV path	1+	3	334

Model number	Arrangement of mediators and moderators	Number of distinct mediators	Number of distinct moderators	Page Number
45	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path, with the other 2 moderating the Mediator-DV path	1+	4	344
46	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path with all 2-way and 3-way interactions, with the other 2 moderating the Mediator-DV path	1+	4	361
47	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path, with the other 2 moderating the Mediator-DV path with all 2-way and 3-way interactions	1+	4	382
48	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path with all 2-way and 3-way interactions, with the other 2 moderating the Mediator-DV path with all 2-way and 3-way interactions	1+	4	402
49	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path, with the other 2 moderating the Mediator-DV path	1+	4	428
50	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path	1+	4	445
51	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path with all 2-way and 3-way interactions, with the other 2 moderating the Mediator-DV path	1+	4	462
52	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path with all 2-way and 3-way interactions, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path	1+	4	484
53	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path, with the other 2 moderating the Mediator-DV path with all 2-	1+	4	505

Model number	Arrangement of mediators and moderators	Number of distinct mediators	Number of distinct moderators	Page Number
	way and 3-way interactions			
54	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path with all 2-way and 3-way interactions	1+	4	526
55	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path with all 2-way and 3-way interactions, with the other 2 moderating the Mediator-DV path with all 2-way and 3-way interactions	1+	4	547
56	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path with all 2-way and 3-way interactions, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path with all 2-way and 3-way interactions	1+	4	574
57	1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path with all 2-way and 3-way interactions, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path with all 2-way and 3-way interactions	1+	4	601
58	1 or more mediators, in parallel if multiple (example uses 1), 1 moderators, which moderates both the IV-Mediator path and the Mediator-DV path	1+	1	633
59	1 or more mediators, in parallel if multiple (example uses 1), 1 moderators, which moderates all of the IV-Mediator path, the Mediator-DV path and the direct IV-DV path	1+	1	638
60	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV-Mediator path, 1 of which also moderates the Mediator-DV path	1+	2	643
61	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV-Mediator path, 1 of which also moderates both the	1+	2	649

Model number	Arrangement of mediators and moderators	Number of distinct mediators	Number of distinct moderators	Page Number
	Mediator-DV path and the direct IV-DV path			
62	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV-Mediator path, 1 of which also moderates the Mediator-DV path, the with the other moderating the direct IV-DV path	1+	2	655
63	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV-Mediator path and the direct IV-DV path, 1 of which also moderates the Mediator-DV path	1+	2	661
64	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the Mediator-DV path, 1 of which also moderates the IV-Mediator path	1+	2	667
65	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the Mediator-DV path, 1 of which also moderates both the IV-Mediator path and the direct IV-DV path	1+	2	673
66	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the Mediator-DV path, 1 of which also moderates the IV-Mediator path, the with the other moderating the direct IV-DV path	1+	2	679
67	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the Mediator-DV path and the direct IV-DV path, 1 of which also moderates the IV-Mediator path	1+	2	685
68	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating the IV-Mediator path with all 2-way and 3-way interactions, 1 of which also moderates the Mediator-DV path	1+	2	691
69	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both of the IV-Mediator path and the direct IV-DV path, with all 2-way and 3-way interactions, 1 of which also moderates the Mediator-DV path	1+	2	697
70	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating the Mediator- DV path with all 2-way and 3-way	1+	2	704

Model number	Arrangement of mediators and moderators	Number of distinct mediators	Number of distinct moderators	Page Number
	interactions, 1 of which also moderates the IV-Mediator path			
71	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both of the Mediator-DV path and the direct IV-DV path, with all 2-way and 3-way interactions, 1 of which also moderates the IV-Mediator path	1+	2	710
72	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV-Mediator path and the Mediator-DV path, with all 2-way and 3-way interactions	1+	2	717
73	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating each of the IV-Mediator path, the Mediator-DV path and the direct IV-DV path, with all 2-way and 3-way interactions	1+	2	724
74	1 or more mediators, in parallel if multiple (example uses 1), IV also moderates the Mediator-DV path	1+	(1)	732
75	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV-Mediator path and the Mediator-DV path	1+	2	736
76	1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating each of the IV-Mediator path, the Mediator-DV path and the direct IV-DV path	1+	2	742
77	3 or more mediators, both in parallel and in series	3+	0	749
78	2 or more mediators, in series, 1 moderator moderating path between mediators	2+	1	753
79	3 or more mediators, both in parallel and in series, 2 moderators, 1 moderating paths between predictor and mediator, the second moderating paths between mediators, and between mediator and DV	3+	2	758
80	1 mediator, predictor has non-linear effect on mediator and outcome	1	0	764

[BLANK PAGE]

Model 1a: 1 moderator [BASIC MODERATION]

Continuous moderator

Example Variables: 1 predictor X, 1 moderator W, 1 outcome Y

Preliminary notes:

The code below assumes that:

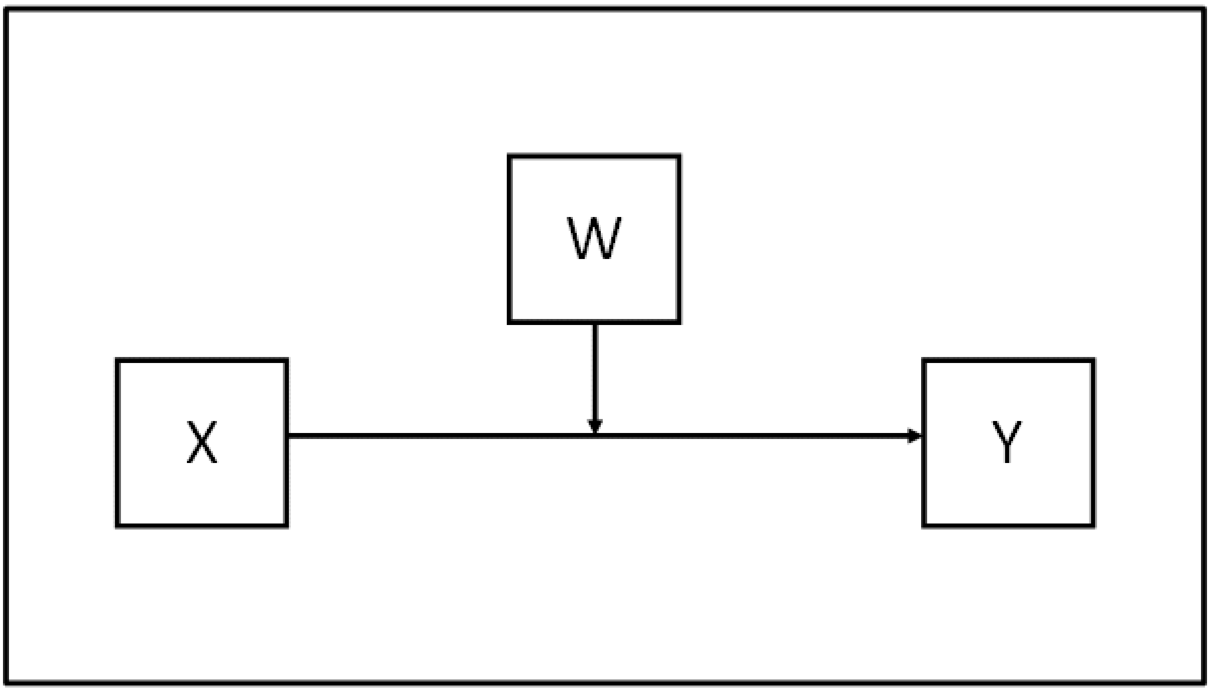
The primary IV (variable X) is continuous or dichotomous

Any moderators (variables W,V,Q,Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for this model at the end of this user guide.

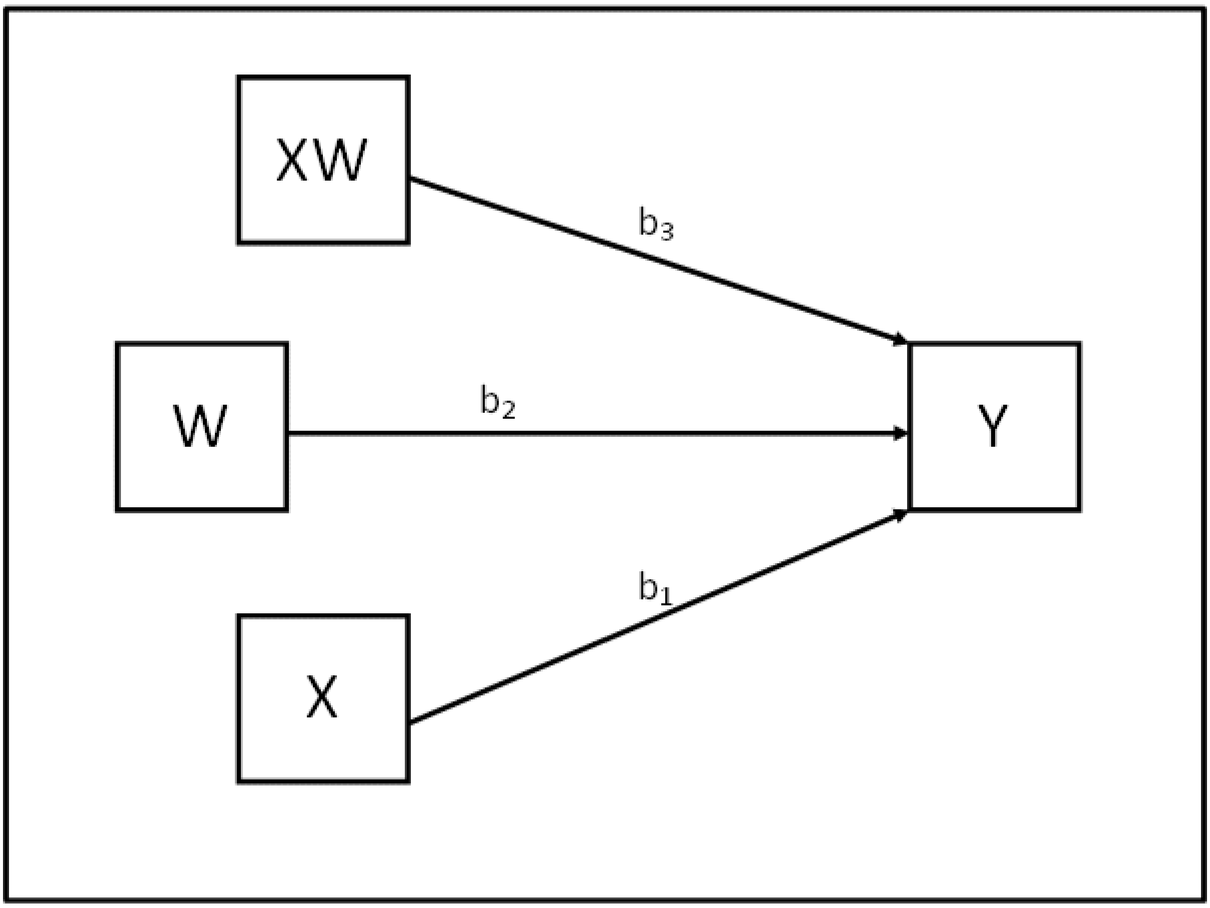
Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous

The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for this model at the bottom of the code (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1X + b2W + b3XW

Albegra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1X + b2W + b3XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + b2W) + (b1 + b3W)X

Hence...

One direct effect of X on Y, conditional on W:

b1 + b3W

M*plus* code for the model:

```
! Predictor variable - X
! Mediator variable(s) – (not applicable)
! Moderator variable(s) - W
! Outcome variable - Y

USEVARIABLES = X W Y XW;

! Create interaction term
! Note that it has to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON X (b1);
   Y ON W (b2);
   Y ON XW (b3);

! Use model constraint subcommand to test simple slopes
! You need to pick low, medium and high moderator values,
! for example, of 1 SD below mean, mean, 1 SD above mean

MODEL CONSTRAINT:
   NEW(LOW_W MED_W HIGH_W SIMP_LO SIMP_MED SIMP_HI);

   LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
   MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
   HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

! Now calc simple slopes for each value of W

   SIMP_LO = b1 + b3*LOW_W;
   SIMP_MED = b1 + b3*MED_W;
   SIMP_HI = b1 + b3*HIGH_W;
```

```
! Use loop plot to plot model for low, med, high values of W
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

   PLOT(LOMOD MEDMOD HIMOD);
   LOOP(XVAL,1,5,0.1);
   LOMOD = (b0 + b2*LOW_W) + (b1 + b3*LOW_W)*XVAL;
   MEDMOD = (b0 + b2*MED_W) + (b1 + b3*MED_W)*XVAL;
   HIMOD = (b0 + b2*HIGH_W) + (b1 + b3*HIGH_W)*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 1b: 1 moderator [BASIC MODERATION], dichotomous moderator

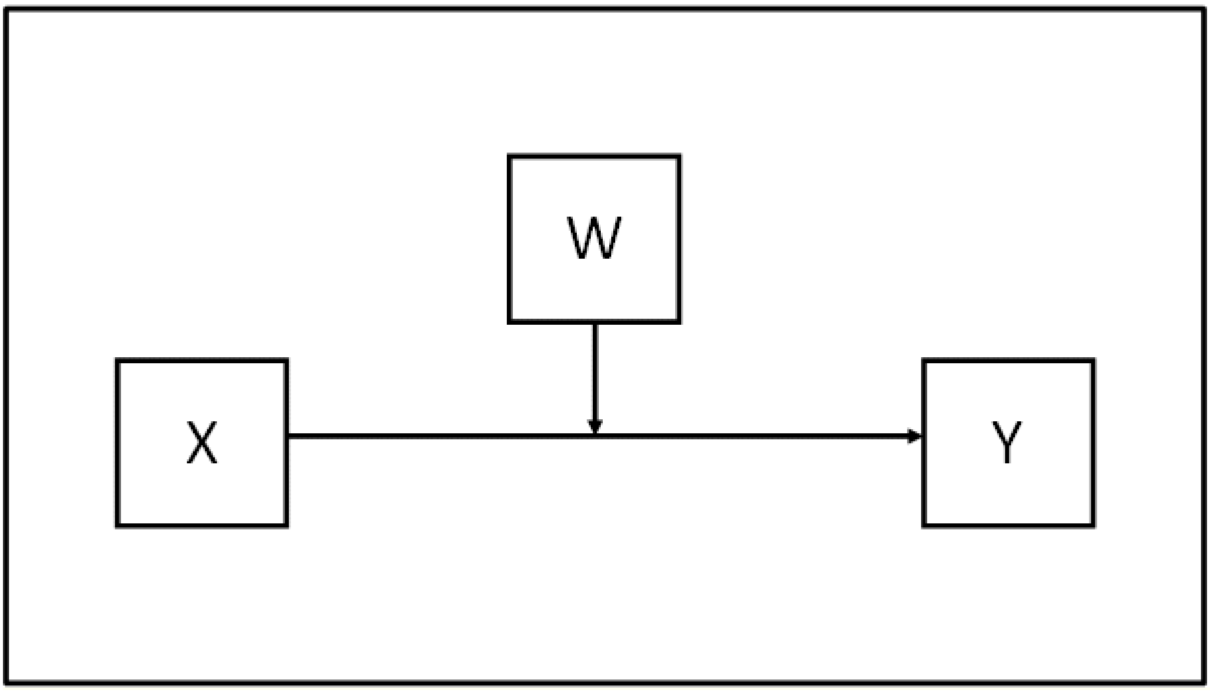
Example Variables: 1 predictor X, 1 moderator W, 1 outcome Y

Preliminary notes:

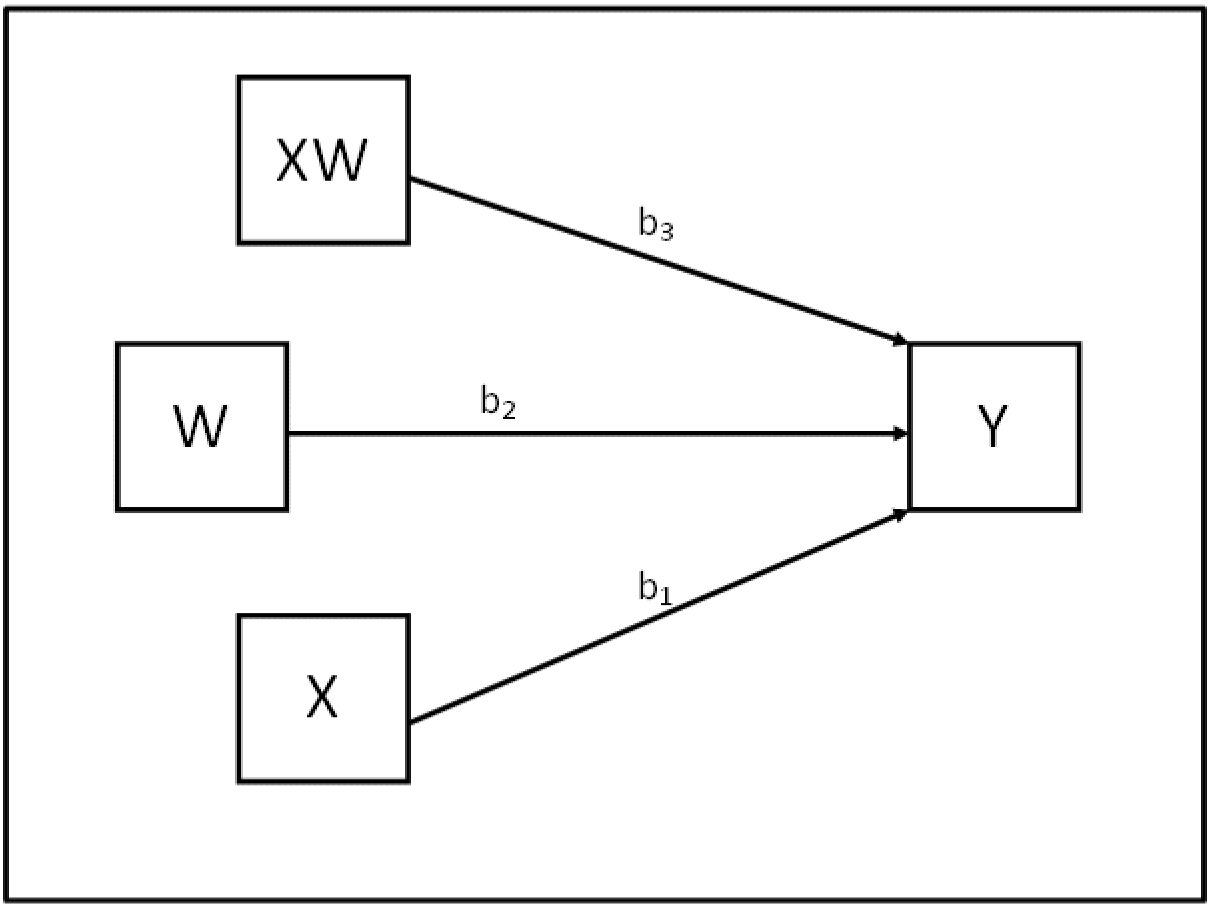
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- The moderator (variable W) is dichotomous. Handling categorical moderators with > 2 categories is demonstrated in model 1d.
- Any mediators (variable M, or M1, M2, etc.) are continuous and satisfy the assumptions of standard multiple regression. An example of how to handle a dichotomous mediator is given in model 4c.
- The DV (variable Y) is continuous and satisfies the assumptions of standard multiple regression. An example of how to handle a dichotomous DV is given in model 1e (i.e. a moderated logistic regression) and in model 4d (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1X + b2W + b3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1X + b2W + b3XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + b2W) + (b1 + b3W)X

Hence...

One direct effect of X on Y, conditional on W:

b1 + b3W

so inserting the values of 0 and 1 for moderator W gives....

when W = 0, Y = b0 + b1X; when W = 1, Y = (b0 + b2) + (b1 + b3)X

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – (not applicable)
! Moderator variable(s) - W, dichotomous, coded 0/1
! Outcome variable - Y

USEVARIABLES = X W Y XW;

! Create interaction term
! Note that it has to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON X (b1);
   Y ON W (b2);
   Y ON XW (b3);

! Use model constraint subcommand to test simple slopes
! You need to insert your two moderator values, 0 and 1

MODEL CONSTRAINT:
   NEW(LOW_W HIGH_W SIMP_LO SIMP_HI);

   LOW_W = 0;
   HIGH_W = 1;

! Now calc simple slopes for each value of W

   SIMP_LO = b1 + b3*LOW_W;
   SIMP_HI = b1 + b3*HIGH_W;

! Use loop plot to plot model for low, med, high values of W
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis
```

```
    PLOT(LOMOD HIMOD);
    LOOP(XVAL,1,5,0.1);
    LOMOD = (b0 + b2*LOW_W) + (b1 + b3*LOW_W)*XVAL;
    HIMOD = (b0 + b2*HIGH_W) + (b1 + b3*HIGH_W)*XVAL;

PLOT:
    TYPE = plot2;

OUTPUT:
    STAND CINT(bcbootstrap);
```

Alternative code:

If you are feeling confident, you could simplify the MODEL CONSTRAINT code to:

```
MODEL CONSTRAINT:
    NEW(SIM_MOD0 SIM_MOD1);
    SIM_MOD0 = b1;
    SIM_MOD1 = b1 + b3;

! Use loop plot to plot model for values of W = 0, W = 1
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(MOD0 MOD1);

    LOOP(XVAL,1,5,0.1);

    MOD0 = b0 + b1*XVAL;
    MOD1 = (b0 + b2) + (b1 + b3)*XVAL;

PLOT:
    TYPE = plot2;

OUTPUT:
    STAND CINT(bcbootstrap);
```

Model 1c: 1 moderator [BASIC MODERATION], dichotomous moderator (using multigroup method)

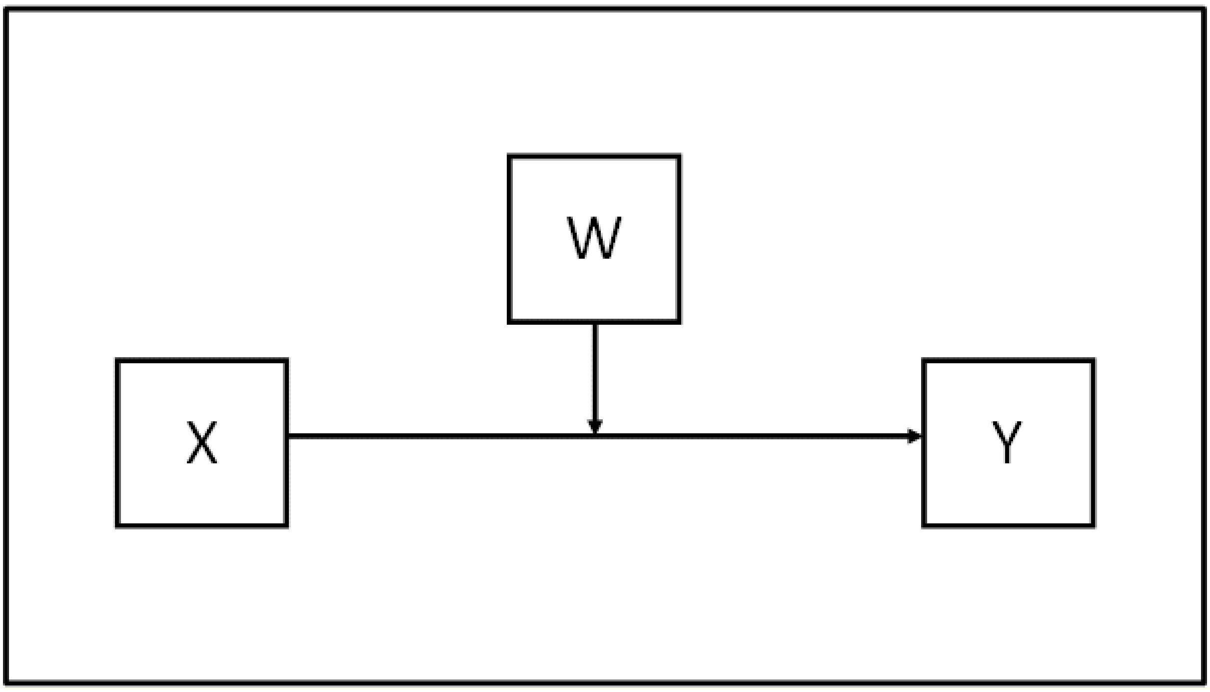
Example Variables: 1 predictor X, 1 moderator W, 1 outcome Y

Preliminary notes:

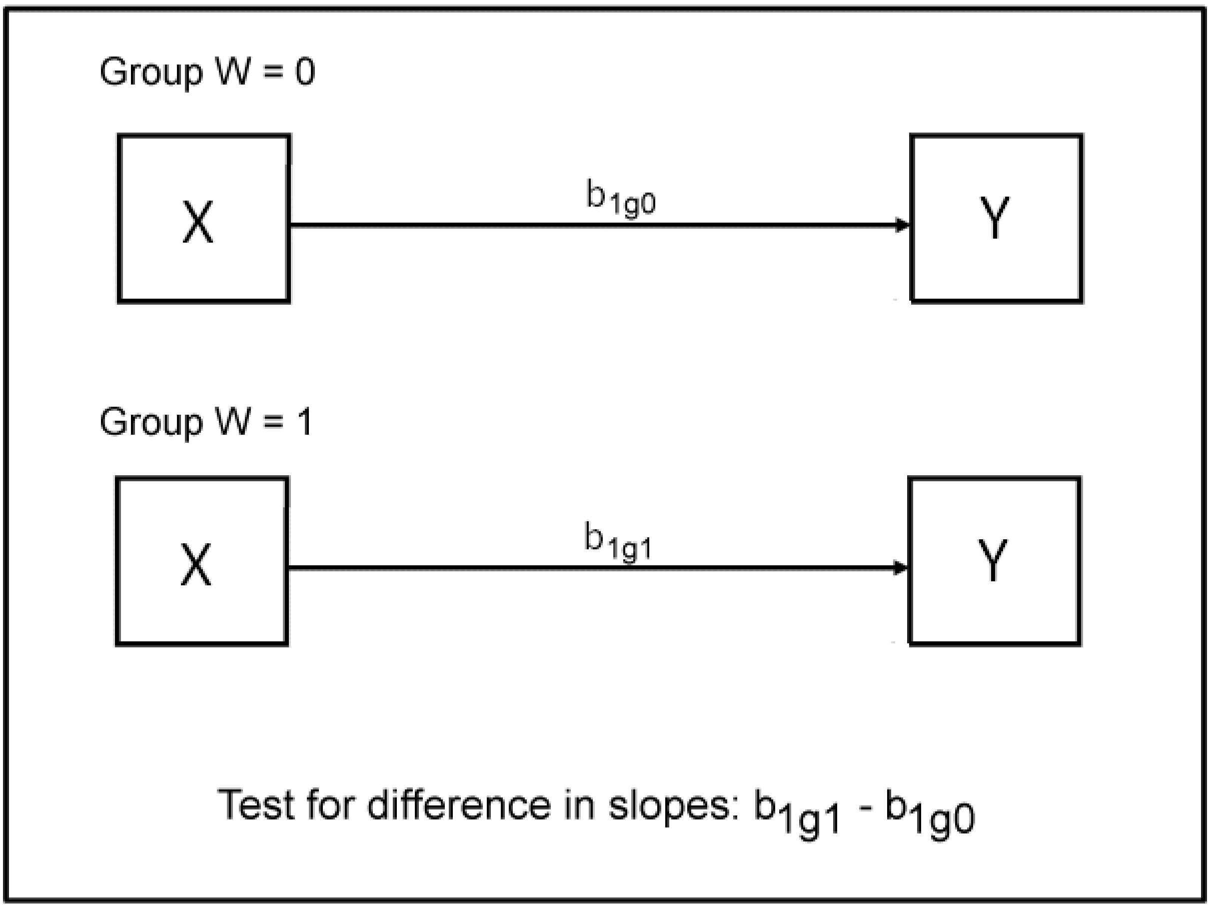
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- The moderator (variable W) is dichotomous. Handling categorical moderators with > 2 categories is demonstrated in model 1d.
- Any mediators (variable M, or M1, M2, etc.) are continuous and satisfy the assumptions of standard multiple regression. An example of how to handle a dichotomous mediator is given in model 4c.
- The DV (variable Y) is continuous and satisfies the assumptions of standard multiple regression. An example of how to handle a dichotomous DV is given in model 1e (i.e. a moderated logistic regression) and in model 4d (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1X + b2W + b3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1X + b2W + b3XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + b2W) + (b1 + b3W)X

Hence...

One direct effect of X on Y, conditional on W:

b1 + b3W

so inserting the values of 0 and 1 for moderator W gives....

when W = 0, Y = b0 + b1X; when W = 1, Y = (b0 + b2) + (b1 + b3)X

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – (not applicable)
! Moderator variable(s) - W, dichotomous, coded 0/1
! Outcome variable - Y

USEVARIABLES = X W Y XW;

! Define groups of moderator W

GROUPING = W (0 = GP0 1 = GP1);

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement first state basic regression that is
being moderated

MODEL:
   Y ON X;

! Then restate for each group, naming each group's intercept
and slope coefficient
! and fixing residual variances equal

  MODEL GP0:
   [Y](b0g0);
   Y ON X (b1g0);
   Y (vary);

  MODEL GP1:
   [Y](b0g1);
   Y ON X (b1g1);
   Y (vary);

! Use model constraint subcommand to create and test
difference in slopes
! Note that slopes for each group provide simple slopes tests
already
```

```
MODEL CONSTRAINT:
   NEW(b3);

   b3 = b1g1 - b1g0;

! Use loop plot to plot model for values of W = 0, W = 1
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

   PLOT(LINEGP0 LINEGP1);
   LOOP(XVAL,1,5,0.1);

   LINEGP0 = b0g0 + b1g0*XVAL;
   LINEGP1 = b0g1 + b1g1*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 1d: 1 moderator [BASIC MODERATION], categorical moderator with > 2 categories

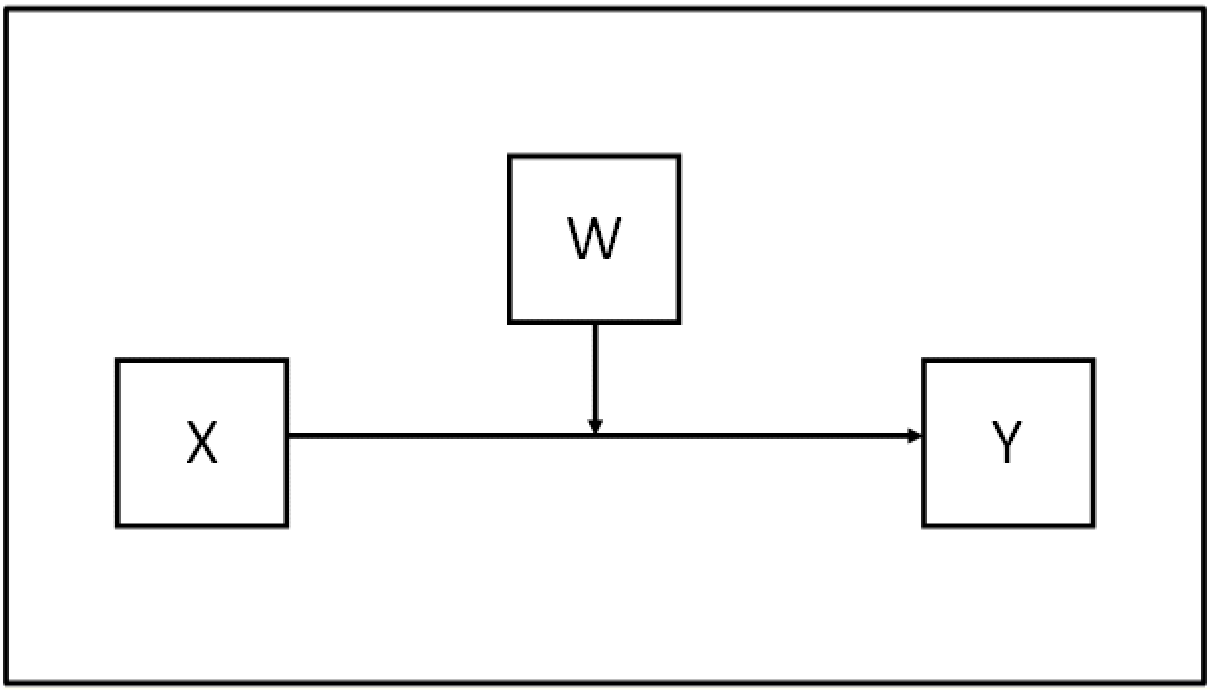
Example Variables: 1 predictor X, 2 dummy variables WD1 and WD2 representing 3 category moderator W, 1 outcome Y

Preliminary notes:

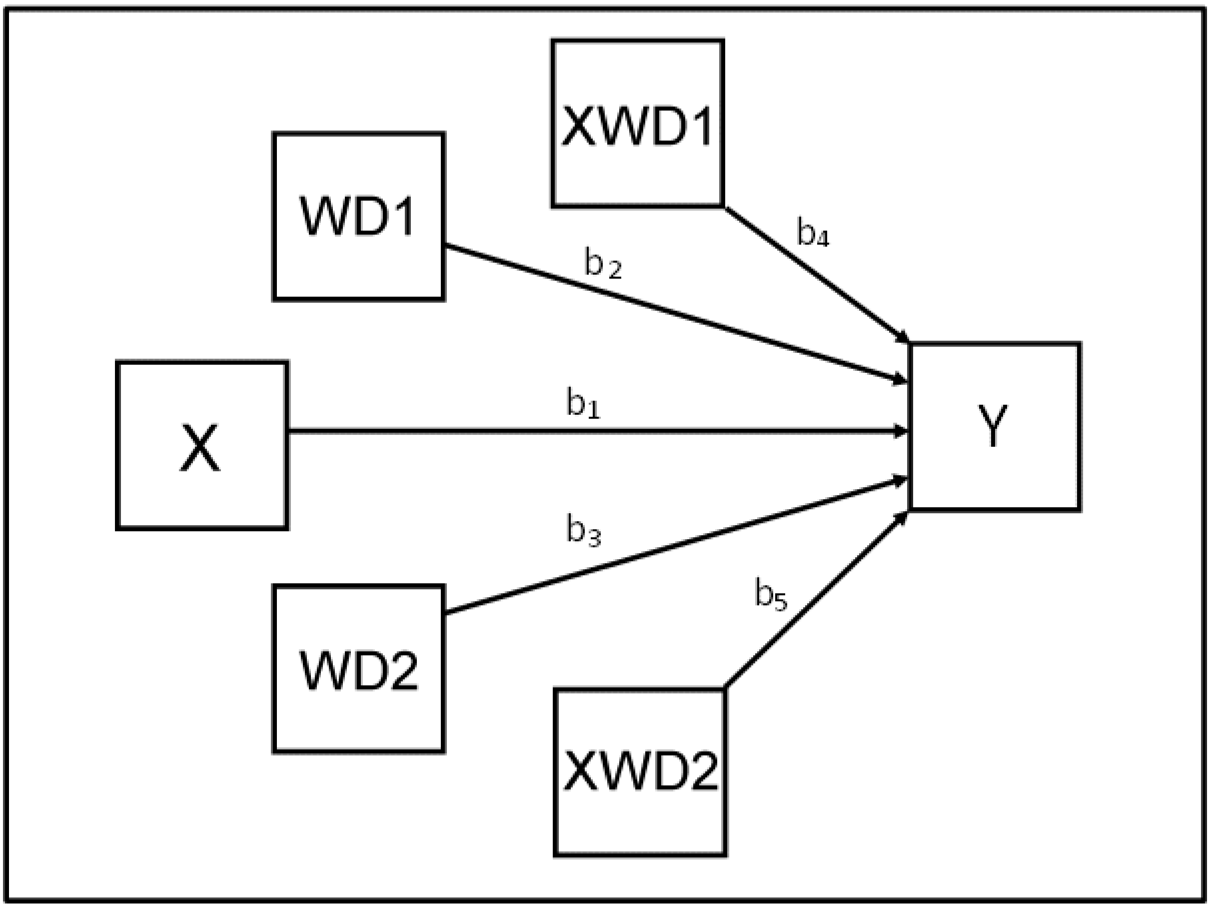
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- The moderator (variable W) has 3 categories (1-3), and is therefore represented by two dummy variables WD1, WD2, coded such that when W = 1, then WD1 = 1, WD2 = 0; when W = 2, then WD1 = 0, WD2 = 1; when W = 3 (the reference category), then WD1 = 0, WD2 = 0.
- Any mediators (variable M, or M1, M2, etc.) are continuous and satisfy the assumptions of standard multiple regression. An example of how to handle a dichotomous mediator is given in model 4c.
- The DV (variable Y) is continuous and satisfies the assumptions of standard multiple regression. An example of how to handle a dichotomous DV is given in model 1e (i.e. a moderated logistic regression) and in model 4d (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1X + b2WD1 + b3WD2 + b4XWD1 + b5XWD2

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1X + b2WD1 + b3WD2 + b4XWD1 + b5XWD2

Hence... grouping terms into form Y = a + bX

Y = (b0 + b2WD1 + b3WD2) + (b1 + b4WD1 + b5WD2)X

Hence...

One direct effect of X on Y, conditional on W:

b1 + b4WD1 + b5WD2

so inserting the values of 0 and 1 for moderator W gives....

when W = 1, then WD1 = 1, WD2 = 0, hence Y = (b0 + b2) + (b1 + b4)X
when W = 2, then WD1 = 0, WD2 = 1, hence Y = (b0 + b3) + (b1 + b5)X
when W = 3, then WD1 = 0, WD2 = 0, hence Y = b0 + b1X

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – (not applicable)
! Moderator variable(s) - W, 3 categories, represented by
dichotomous 0/1 dummy variables WD1, WD2
! Outcome variable - Y

USEVARIABLES = X WD1 WD2 Y XWD1 XWD2;

! Create interaction term
! Note that it has to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XWD1 = X*WD1;
   XWD2 = X*WD2;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON X (b1);
   Y ON WD1 (b2);
   Y ON WD2 (b3);
   Y ON XWD1 (b4);
   Y ON XWD2 (b5);

! Use model constraint subcommand to test simple slopes
! You need to insert your respective dummy variable values, 0
and 1, for each group of W

MODEL CONSTRAINT:
   NEW(SIMP_W1 SIMP_W2 SIMP_W3);

! Now calc simple slopes for each group of W

   SIMP_W1 = b1 + b4;
   SIMP_W2 = b1 + b5;
   SIMP_W3 = b1;
```

```
! Use loop plot to plot model for values of W = 0, W = 1
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

   PLOT(LINE_W1 LINE_W2 LINE_W3);
   LOOP(XVAL,1,5,0.1);

   LINE_W1 = (b0 + b2) + (b1 + b4)*XVAL;
   LINE_W2 = (b0 + b3) + (b1 + b5)*XVAL;
   LINE_W3 = b0 + b1*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 1e: 1 moderator [BASIC MODERATION], dichotomous outcome (logistic regression)

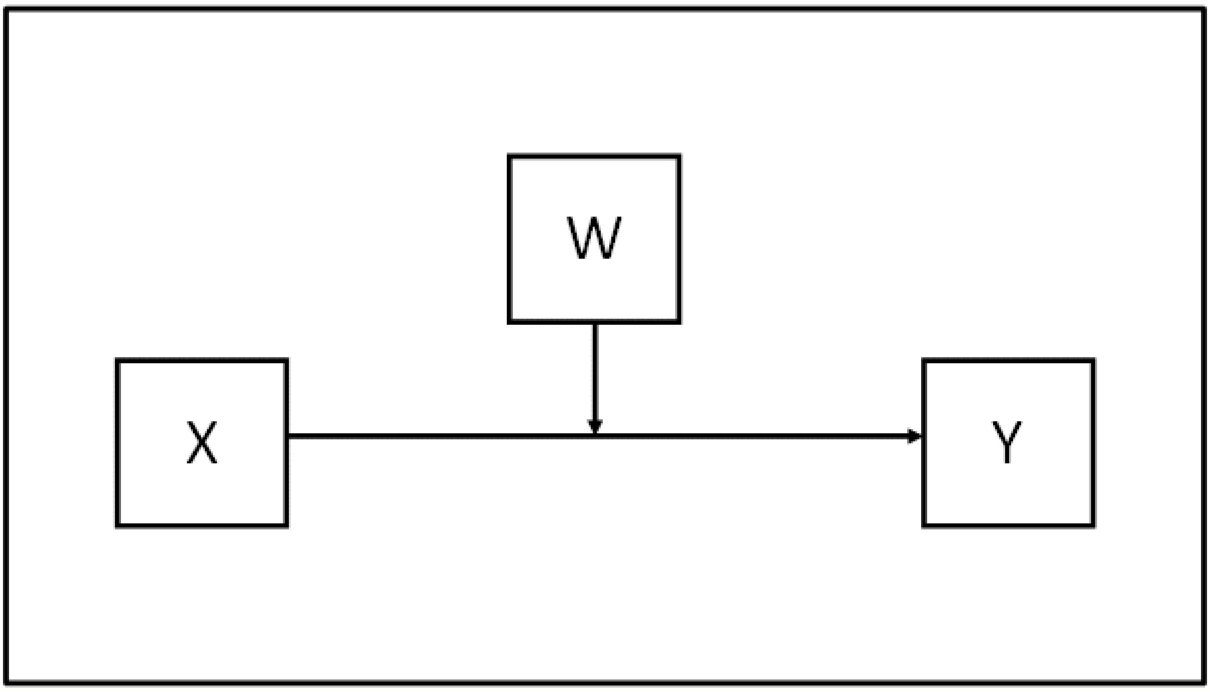
Example Variables: 1 predictor X, 1 moderator W, 1 outcome Y

Preliminary notes:

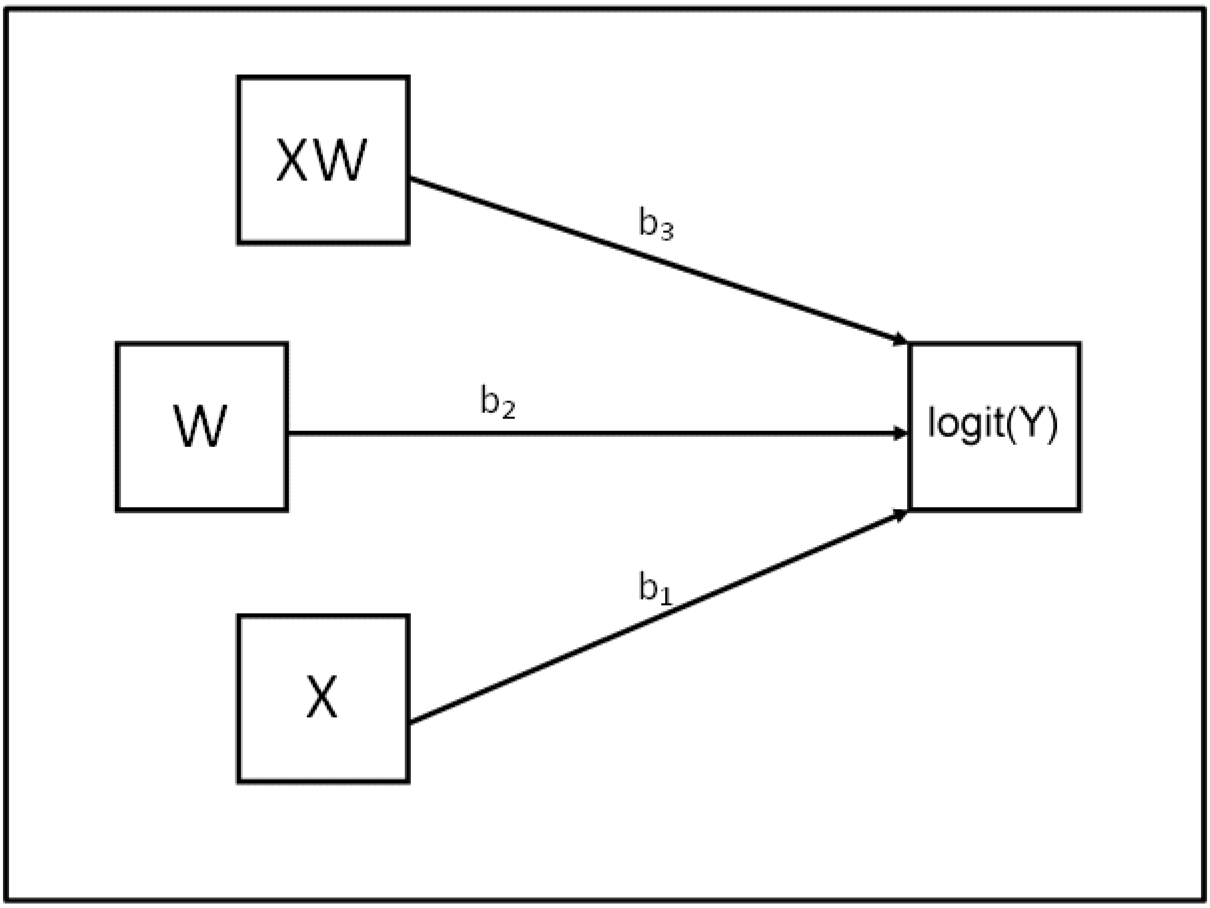
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given in model 1b. Handling categorical moderators with > 2 categories is demonstrated in model 1d.
- Any mediators (variable M, or M1, M2, etc.) are continuous and satisfy the assumptions of standard multiple regression. An example of how to handle a dichotomous mediator is given in model 4c.
- The DV (variable Y) is dichotomous and satisfies the assumptions of logistic regression.

Model Diagram:



Statistical Diagram:



Model Equation(s):

logit(Y) = b0 + b1X + b2W + b3XW

Algebra to calculate indirect and/or conditional effects by writing model as logit(Y) = a + bX:

logit(Y) = b0 + b1X + b2W + b3XW

Hence... grouping terms into form logit(Y) = a + bX

logit(Y) = (b0 + b2W) + (b1 + b3W)X

Hence...

One direct effect of X on logit(Y), conditional on W:

b1 + b3W

Hence, writing as an odds ratio...

The multiplicative effect of X on the odds of Y, conditional on W:

exp(b1 + b3W) = exp(b1)*exp(b3W)

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – (not applicable)
! Moderator variable(s) - W
! Outcome variable - Y - a dichotomous outcome, coded 0/1

USEVARIABLES = X W Y XW;

CATEGORICAL = Y;

! Create interaction term
! Note that it has to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y$1] (b0);
   Y ON X (b1);
   Y ON W (b2);
   Y ON XW (b3);

! Use model constraint subcommand to test simple slopes
! You need to pick low, medium and high moderator values,
! for example, of 1 SD below mean, mean, 1 SD above mean

MODEL CONSTRAINT:
   NEW(LOW_W MED_W HIGH_W OR_LO OR_MED OR_HI);

   LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
   MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
   HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

! Now calc conditional odds ratios for each value of W
```

```
    OR_LO = exp(b1 + b3*LOW_W);
    OR_MED = exp(b1 + b3*MED_W);
    OR_HI = exp(b1 + b3*HIGH_W);

! Use loop plot to plot predicted probabilities by X
! conditional on low, med, high values of W
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOMOD PMEDMOD PHIMOD);
    LOOP(XVAL,1,5,0.1);
    PLOMOD = 1/(1 + exp(-1*((b0 + b2*LOW_W) + (b1 +
b3*LOW_W)*XVAL)));
    PMEDMOD = 1/(1 + exp(-1*((b0 + b2*MED_W) + (b1 +
b3*MED_W)*XVAL)));
    PHIMOD = 1/(1 + exp(-1*((b0 + b2*HIGH_W) + (b1 +
b3*HIGH_W)*XVAL)));

PLOT:
    TYPE = plot2;

OUTPUT:
    STAND;
```

Model 2: 2 moderators, 2-way interactions with predictor only

Example Variables: 1 predictor X, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

The code below assumes that

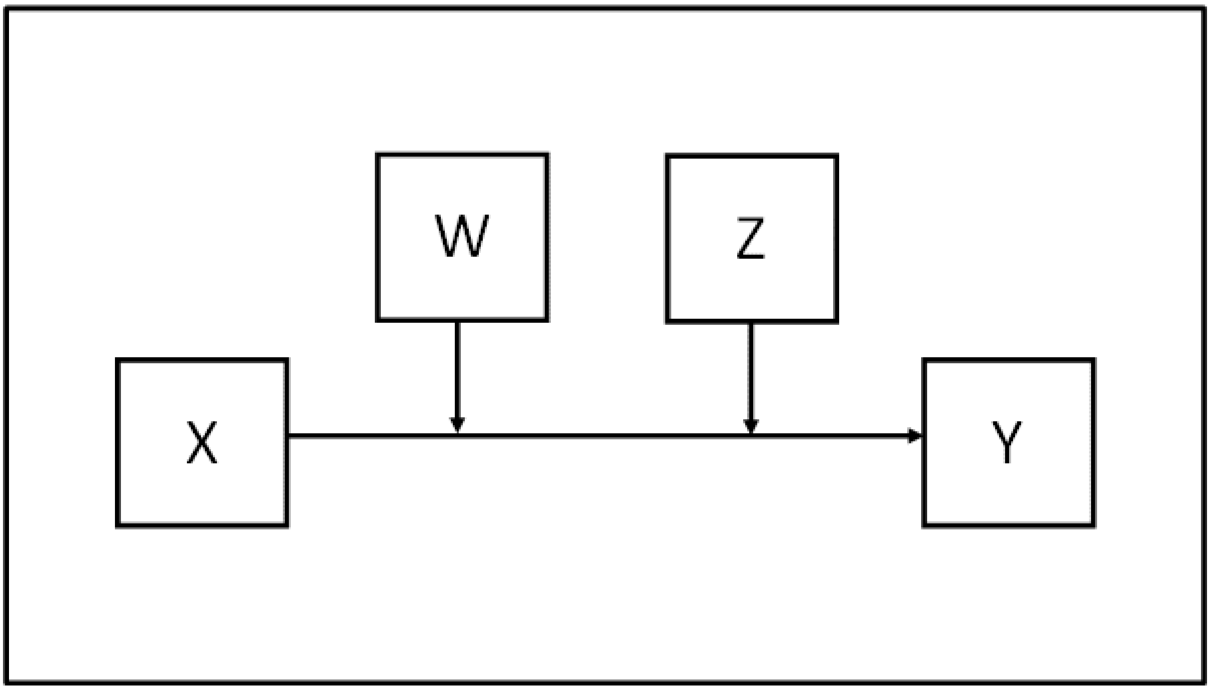
The primary IV (variable X) is continuous or dichotomous

Any moderators (variables W,V,Q,Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation)

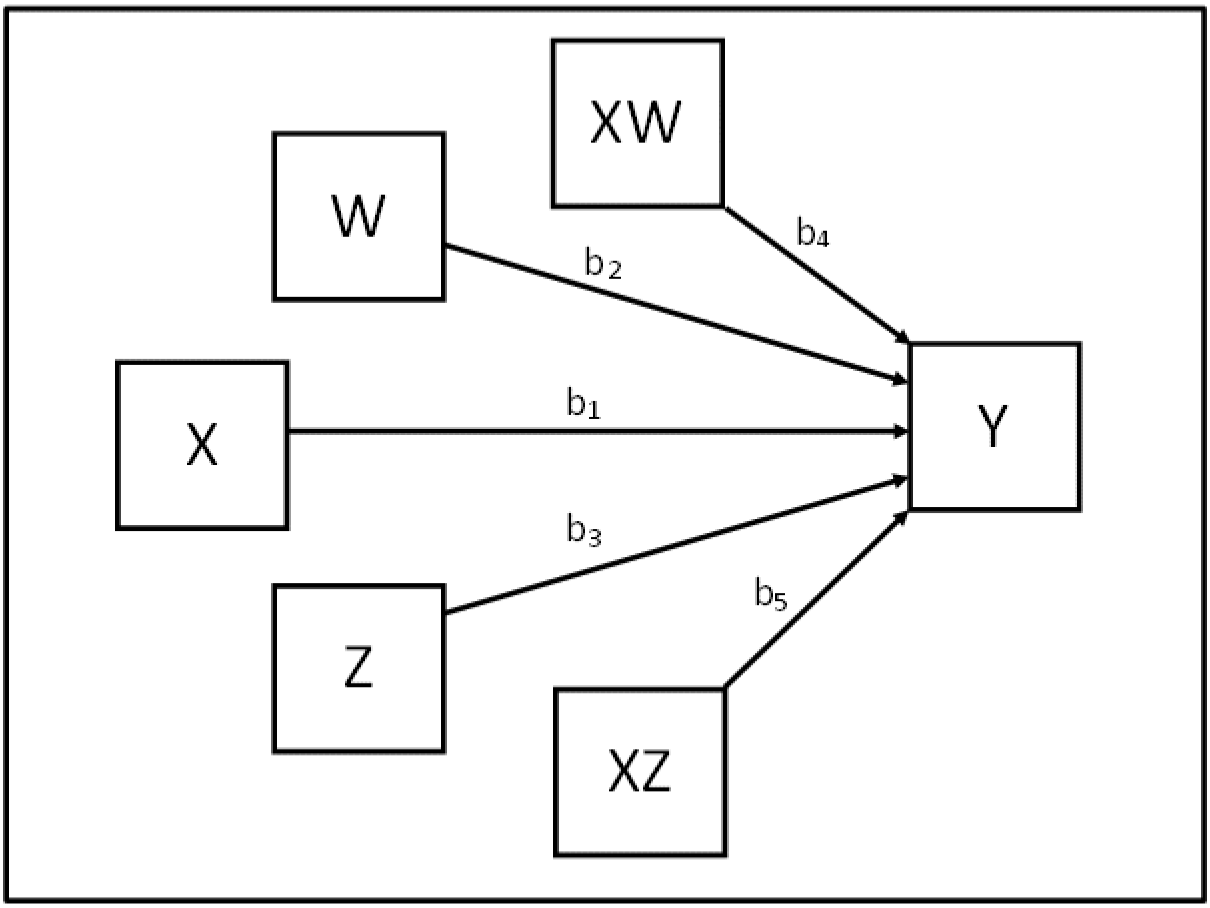
Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous

The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1X + b2W + + b3Z + b4XW + b5XZ

Albegra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1X + b2W + b3Z + b4XW + b5XZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + b2W + b3Z) + (b1 + b4W + b5Z)X

Hence...

One direct effect of X on Y, conditional on W and Z:

b1 + b4W + b5Z

M*plus* code for the model:

```
! Predictor variable - X
! Mediator variable(s) – (not applicable)
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X W Y XW XZ;

! Create interaction terms
! Note as new vars they have to be placed at end of
USEVARIABLES subcommand above

DEFINE:
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path, and intercept, using
parentheses

MODEL:
   [Y] (b0);
   Y ON X (b1);
   Y ON W (b2);
   Y ON Z (b3);
   Y ON XW (b4);
   Y ON XZ (b5);

! Use model constraint subcommand to test simple slopes
! You need to pick low, medium and high moderator values for
both W and Z,
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for simple slopes used below:
! MEW_LOZ = medium value of W and low value of Z, etc.

MODEL CONSTRAINT:
   NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_LOZ MEW_LOZ
HIW_LOZ
   LOW_MEZ MEW_MEZ HIW_MEZ LOW_HIZ MEW_HIZ HIW_HIZ);

   LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
   MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
```

```
   HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

   LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
   MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
   HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

! Now calc simple slopes for each value of W and Z

   LOW_LOZ = b1 + b4*LOW_W + b5*LOW_Z;
   MEW_LOZ = b1 + b4*MED_W + b5*LOW_Z;
   HIW_LOZ = b1 + b4*HIGH_W + b5*LOW_Z;

   LOW_MEZ = b1 + b4*LOW_W + b5*MED_Z;
   MEW_MEZ = b1 + b4*MED_W + b5*MED_Z;
   HIW_MEZ = b1 + b4*HIGH_W + b5*MED_Z;

   LOW_HIZ = b1 + b4*LOW_W + b5*HIGH_Z;
   MEW_HIZ = b1 + b4*MED_W + b5*HIGH_Z;
   HIW_HIZ = b1 + b4*HIGH_W + b5*HIGH_Z;

! Use loop plot to plot model for all combinations of low, med,
high values of W and Z
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

   PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ PHIW_MEZ
   PLOW_HIZ PMEW_HIZ PHIW_HIZ);

   LOOP(XVAL,1,5,0.1);

   PLOW_LOZ = (b0 + b2*LOW_W + b3*LOW_Z) + LOW_LOZ*XVAL;
   PMEW_LOZ = (b0 + b2*MED_W + b3*LOW_Z) + MEW_LOZ*XVAL;
   PHIW_LOZ = (b0 + b2*HIGH_W + b3*LOW_Z) + HIW_LOZ*XVAL;

   PLOW_MEZ = (b0 + b2*LOW_W + b3*MED_Z) + LOW_MEZ*XVAL;
   PMEW_MEZ = (b0 + b2*MED_W + b3*MED_Z) + MEW_MEZ*XVAL;
   PHIW_MEZ = (b0 + b2*HIGH_W + b3*MED_Z) + HIW_MEZ*XVAL;

   PLOW_HIZ = (b0 + b2*LOW_W + b3*HIGH_Z) + LOW_HIZ*XVAL;
   PMEW_HIZ = (b0 + b2*MED_W + b3*HIGH_Z) + MEW_HIZ*XVAL;
   PHIW_HIZ = (b0 + b2*HIGH_W + b3*HIGH_Z) + HIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 3: 2 moderators, all 2-way and 3-way interactions

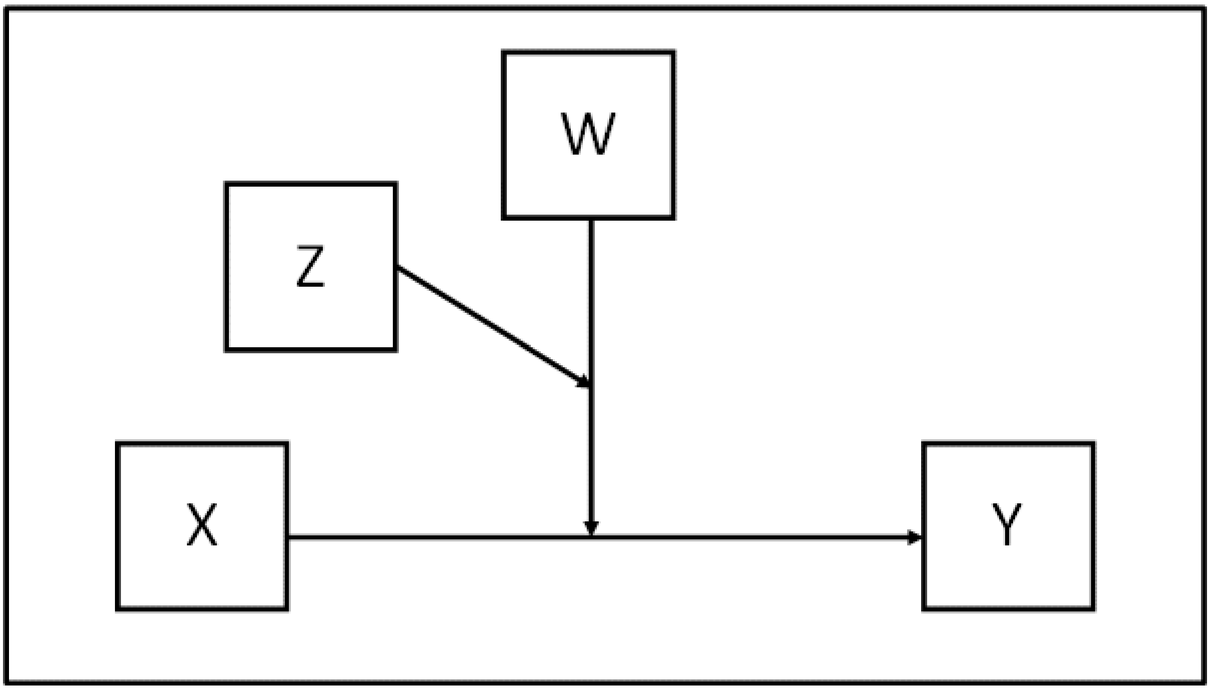
Example Variables: 1 predictor X, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

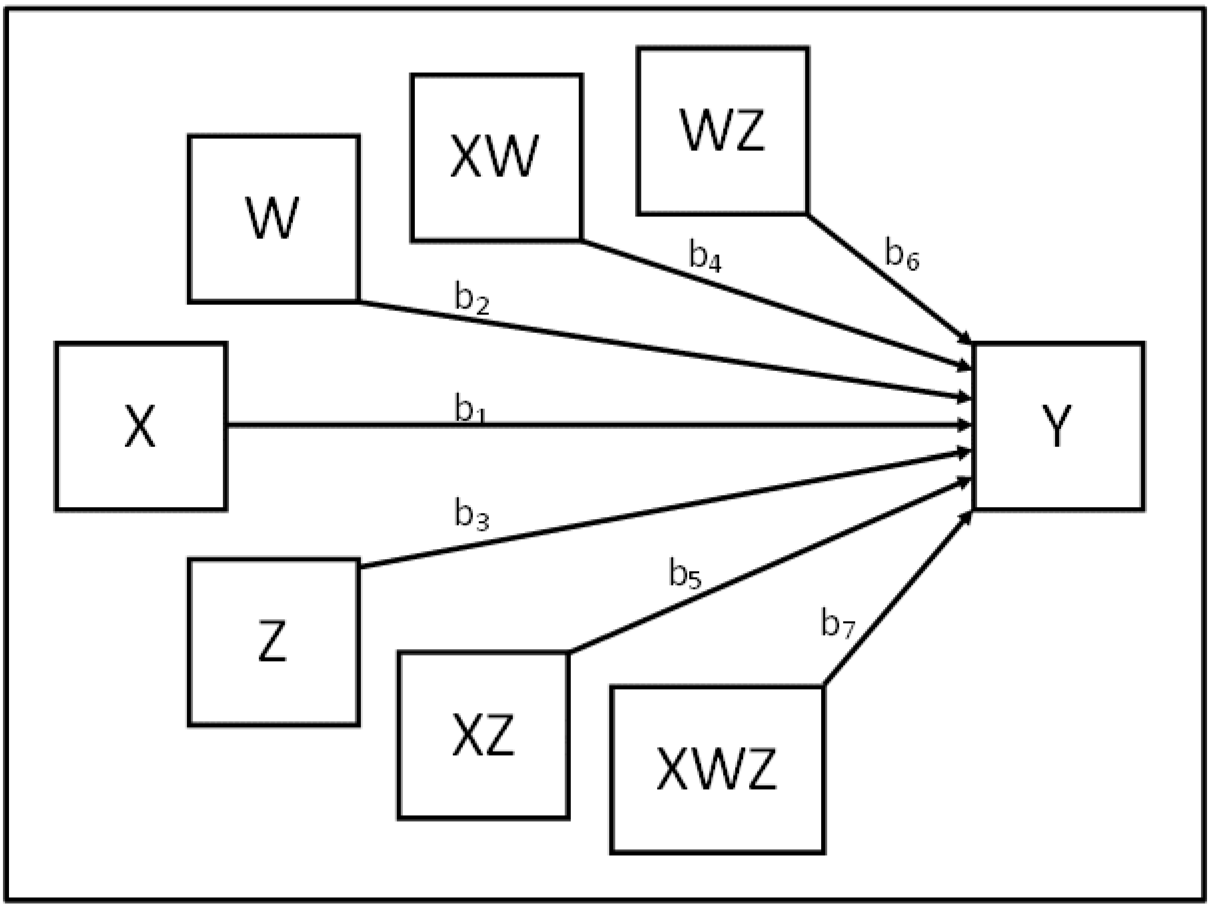
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- Any moderators (variables W,V,Q,Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation)
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1X + b2W + + b3Z + b4XW + b5XZ + b6WZ + b7XWZ

Albegra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1X + b2W + b3Z + b4XW + b5XZ + b6WZ + b7XWZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + b2W + b3Z + b6WZ) + (b1 + b4W + b5Z + b7WZ)X

Hence...

One direct effect of X on Y, conditional on W and Z:

b1 + b4W + b5Z + b7WZ

M*plus* code for the model:

```
! Predictor variable - X
! Mediator variable(s) – (not applicable)
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X W Y XW XZ WZ XWZ;

! Create interaction terms
! Note that as new vars they have to be placed at end of
USEVARIABLES subcommand above

DEFINE:
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path, and intercept, using
parentheses

MODEL:
   [Y] (b0);
   Y ON X (b1);
   Y ON W (b2);
   Y ON Z (b3);
   Y ON XW (b4);
   Y ON XZ (b5);
   Y ON WZ (b6);
   Y ON XWZ (b7);

! Use model constraint subcommand to test simple slopes
! You need to pick low, medium and high moderator values for
both W and Z,
! for example, of 1 SD below mean, mean, 1 SD above mean
! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention used below: MEW_LOZ = medium
value of W and low value of Z, etc.

MODEL CONSTRAINT:
   NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_LOZ MEW_LOZ
HIW_LOZ
   LOW_MEZ MEW_MEZ HIW_MEZ LOW_HIZ MEW_HIZ HIW_HIZ);

   LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
```

```
   MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
   HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

   LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
   MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
   HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

! Now calc simple slopes for each value of W and Z

   LOW_LOZ = b1 + b4*LOW_W + b5*LOW_Z + b7*LOW_W*LOW_Z;
   MEW_LOZ = b1 + b4*MED_W + b5*LOW_Z + b7*MED_W*LOW_Z;
   HIW_LOZ = b1 + b4*HIGH_W + b5*LOW_Z + b7*HIGH_W*LOW_Z;

   LOW_MEZ = b1 + b4*LOW_W + b5*MED_Z + b7*LOW_W*MED_Z;
   MEW_MEZ = b1 + b4*MED_W + b5*MED_Z + b7*MED_W*MED_Z;
   HIW_MEZ = b1 + b4*HIGH_W + b5*MED_Z + b7*HIGH_W*MED_Z;

   LOW_HIZ = b1 + b4*LOW_W + b5*HIGH_Z + b7*LOW_W*HIGH_Z;
   MEW_HIZ = b1 + b4*MED_W + b5*HIGH_Z + b7*MED_W*HIGH_Z;
   HIW_HIZ = b1 + b4*HIGH_W + b5*HIGH_Z + b7*HIGH_W*HIGH_Z;

! Use loop plot to plot model for all combinations of low, med,
high values of W and Z
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

   PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ PHIW_MEZ
   PLOW_HIZ PMEW_HIZ PHIW_HIZ);

   LOOP(XVAL,1,5,0.1);

   PLOW_LOZ = (b0 + b2*LOW_W + b3*LOW_Z + b6*LOW_W*LOW_Z) +
LOW_LOZ*XVAL;
   PMEW_LOZ = (b0 + b2*MED_W + b3*LOW_Z + b6*MED_W*LOW_Z) +
MEW_LOZ*XVAL;
   PHIW_LOZ = (b0 + b2*HIGH_W + b3*LOW_Z + b6*HIGH_W*LOW_Z) +
HIW_LOZ*XVAL;

   PLOW_MEZ = (b0 + b2*LOW_W + b3*MED_Z + b6*LOW_W*MED_Z) +
LOW_MEZ*XVAL;
   PMEW_MEZ = (b0 + b2*MED_W + b3*MED_Z + b6*MED_W*MED_Z) +
MEW_MEZ*XVAL;
   PHIW_MEZ = (b0 + b2*HIGH_W + b3*MED_Z + b6*HIGH_W*MED_Z) +
HIW_MEZ*XVAL;
```

```
   PLOW_HIZ = (b0 + b2*LOW_W + b3*HIGH_Z + b6*LOW_W*HIGH_Z) +
LOW_HIZ*XVAL;
   PMEW_HIZ = (b0 + b2*MED_W + b3*HIGH_Z + b6*MED_W*HIGH_Z) +
MEW_HIZ*XVAL;
   PHIW_HIZ = (b0 + b2*HIGH_W + b3*HIGH_Z + b6*HIGH_W*HIGH_Z)
+ HIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 4a: 1 mediator [BASIC MEDIATION]

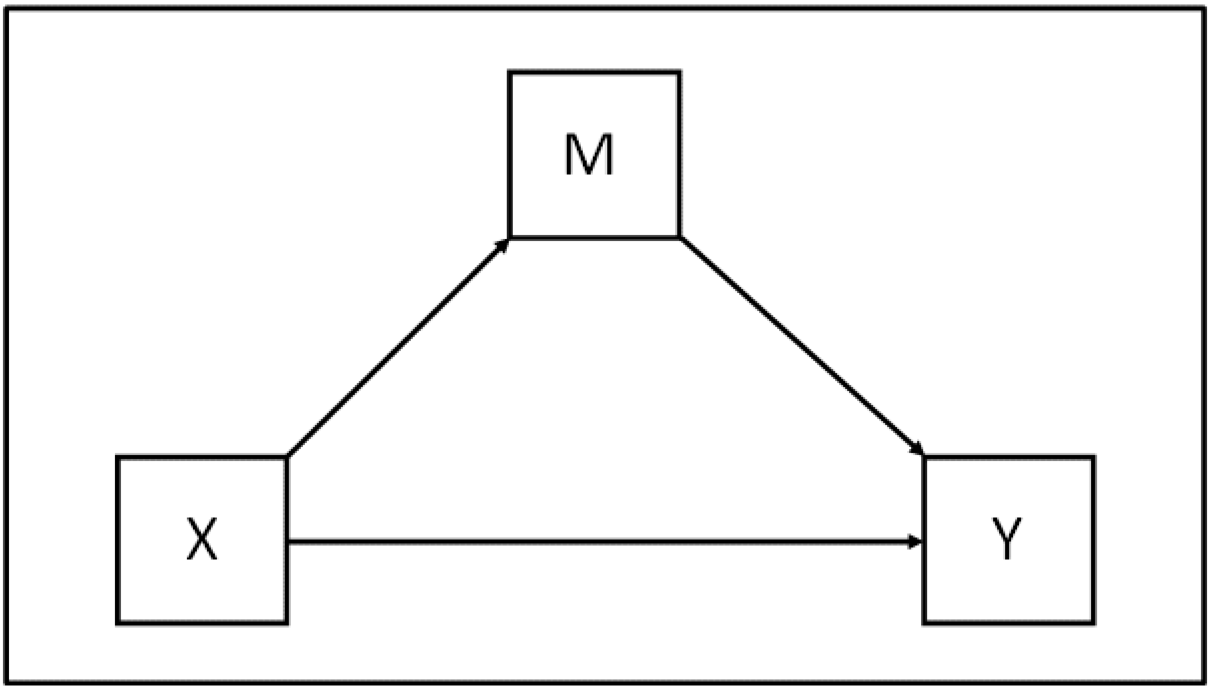
Example Variables: 1 predictor X, 1 mediator M, 1 outcome Y

Preliminary notes:

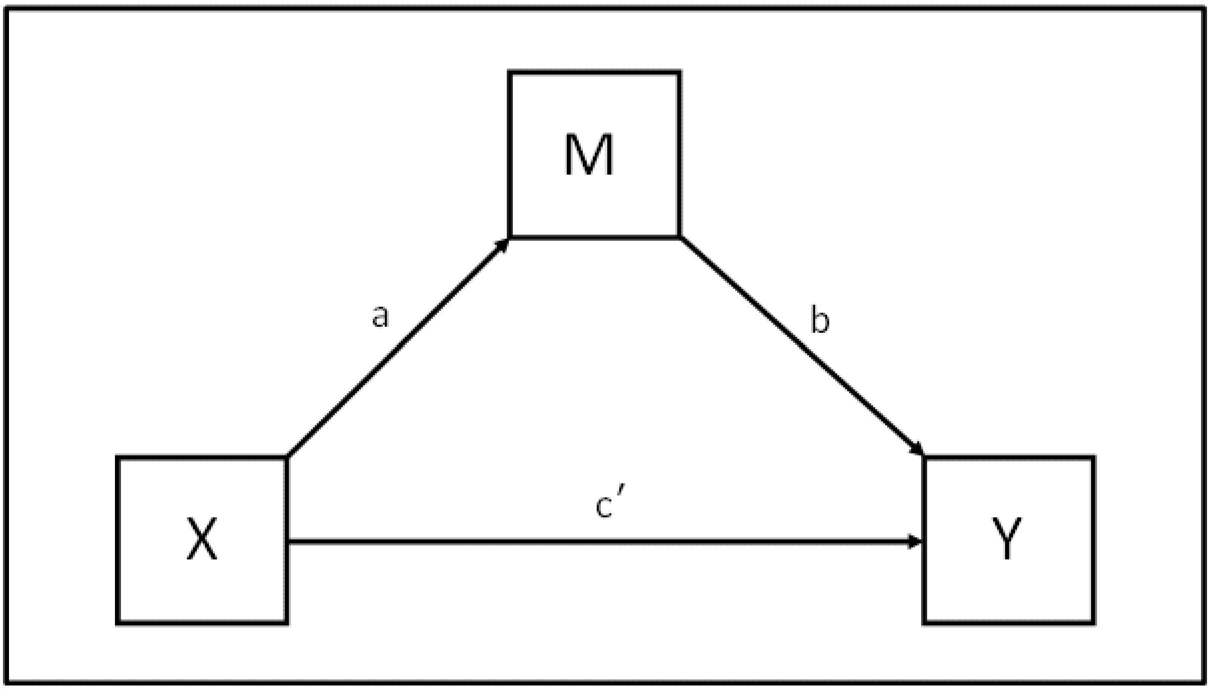
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- The mediator (variable M) is continuous. An example of how to handle a dichotomous mediator is given in model 4c.
- The DV (variable Y) is continuous and satisfies the assumptions of standard multiple regression. An example of how to handle a dichotomous DV is given in model 1e (i.e. a moderated logistic regression) and in model 4d (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c'X
M = a0 + a1X

Algebra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c'X
M = a0 + a1X

Hence... substituting in equations for M

Y = b0 + b1(a0 + a1X) + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1) + (a1b1 + c')X

Hence...

Indirect effect of X on Y:

a1b1

Direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) - none
! Outcome variable - Y

USEVARIABLES = X M Y;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path using parentheses

MODEL:
   Y ON M (b1);

   Y ON X (cdash);    ! direct effect of X on Y

   M ON X (a1);

! Use model constraint to calculate indirect and total effect

MODEL CONSTRAINT:
   NEW(a1b1 TOTAL);
   a1b1 = a1*b1;    ! Indirect effect of X on Y via M
   TOTAL = a1*b1 + cdash;    ! Total effect of X on Y

OUTPUT:
   STAND CINT(bcbootstrap);
```

Editing required for testing indirect effect(s) using alternative MODEL INDIRECT: subcommand:

MODEL INDIRECT: offers an alternative to MODEL CONSTRAINT: for models containing indirect effects, where these are not moderated. To use MODEL INDIRECT: instead, you would edit the code above as follows:

First, you can remove the naming of parameters using parentheses in the MODEL: command, i.e. you just need:

```
MODEL:
   Y ON X M;
   M ON X;
```

Second, replace the MODEL CONSTRAINT: subcommand with the following MODEL INDIRECT: subcommand:

```
   MODEL INDIRECT:
   Y IND X;
```

Leave the OUTPUT: command unchanged.

Model 4b: 2 mediators in parallel [BASIC MEDIATION]

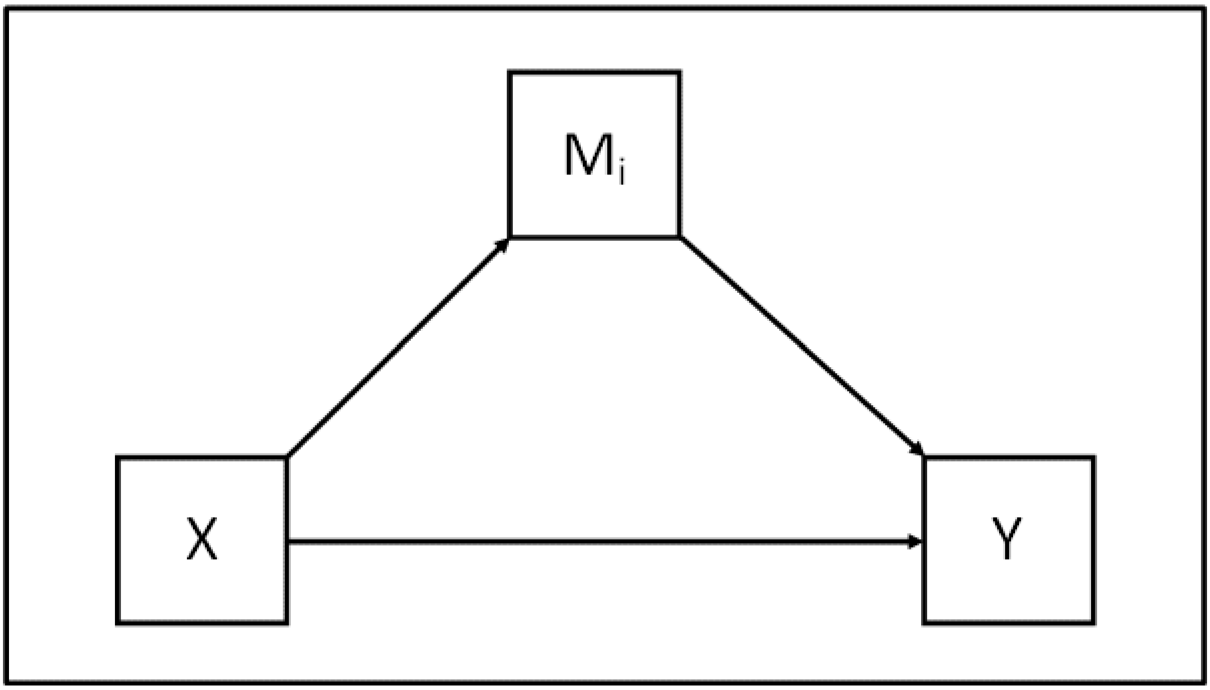
Example Variables: 1 predictor X, 2 mediators M1 and M2, 1 outcome Y

Preliminary notes:

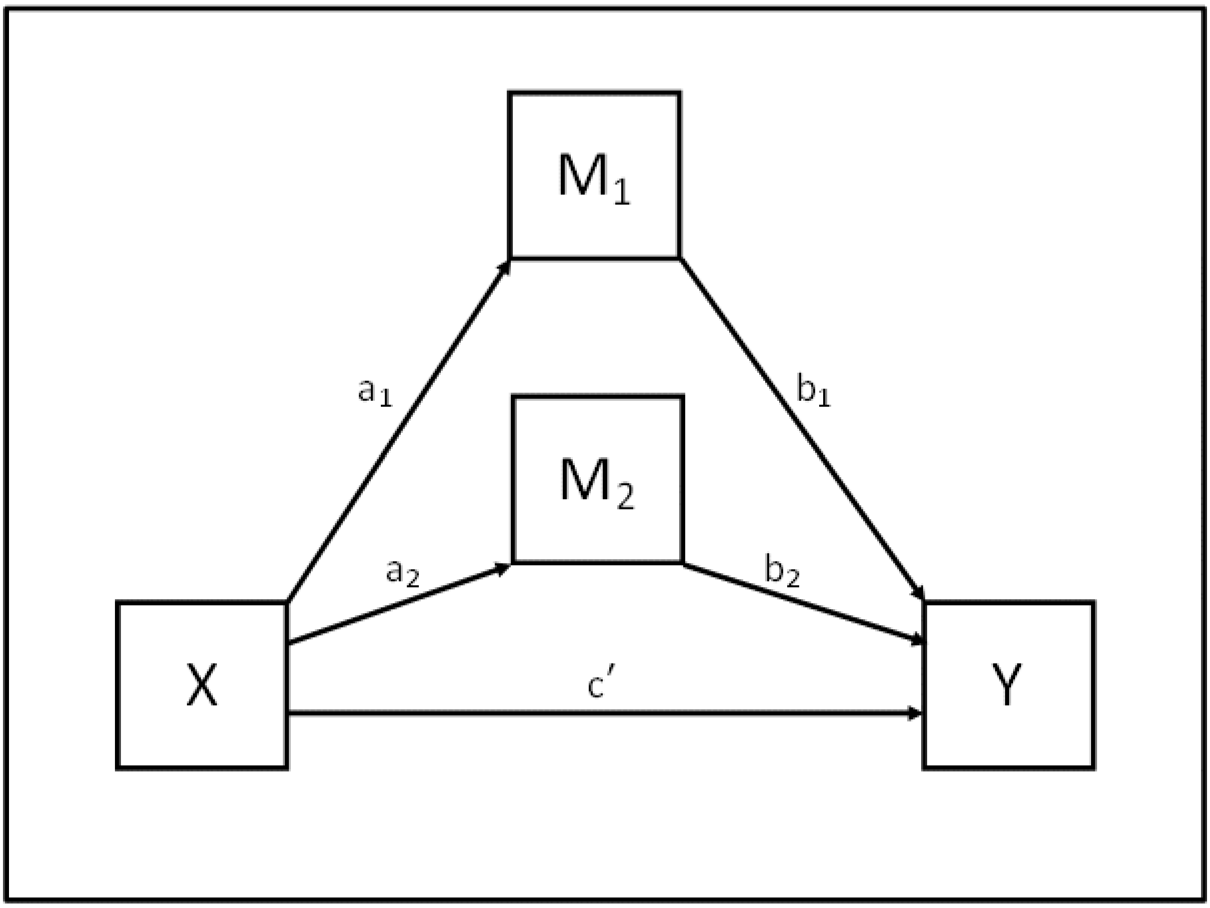
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given in model 1b. Handling categorical moderators with > 2 categories is demonstrated in model 1d.
- Any mediators (variable M, or M1, M2, etc.) are continuous and satisfy the assumptions of standard multiple regression. An example of how to handle a dichotomous mediator is given in model 4c.
- The DV (variable Y) is continuous and satisfies the assumptions of standard multiple regression. An example of how to handle a dichotomous DV is given in model 1e (i.e. a moderated logistic regression) and in model 4d (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M1 + b2M2 + c'X
M1 = a01 + a1X
M2 = a02 + a2X

Algebra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M1 + b2M2 + c'X
M1 = a01 + a1X
M2 = a02 + a2X

Hence... substituting in equations for M1 and M2

Y = b0 + b1(a01 + a1X) + b2(a02 + a2X) + c'X

Hence... multiplying out brackets

Y = b0 + a01b1 + a1b1X + a02b2 + a2b2X + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a01b1 + a02b2) + (a1b1 + a2b2 + c')X

Hence...

Two indirect effects of X on Y:

a1b1, a2b2

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M1, M2
! Moderator variable(s) - none
! Outcome variable - Y

USEVARIABLES = X M1 M2 Y;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path using parentheses

MODEL:
   Y ON M1 (b1);
   Y ON M2 (b2);

   Y ON X (cdash);    ! direct effect of X on Y

   M1 ON X (a1);
   M2 ON X (a2);

! Use model constraint to calculate specific indirect paths
and total indirect effect

MODEL CONSTRAINT:
   NEW(a1b1 a2b2 TOTALIND TOTAL);
   a1b1 = a1*b1;    ! Specific indirect effect of X on Y via
M1
   a2b2 = a2*b2;    ! Specific indirect effect of X on Y via
M2
   TOTALIND = a1*b1 + a2*b2;    ! Total indirect effect of X on
```

```
Y via M1, M2
   TOTAL = a1*b1 + a2*b2 + cdash;    ! Total effect of X on Y

OUTPUT:
   STAND CINT(bcbootstrap);
```

Editing required for testing indirect effect(s) using alternative MODEL INDIRECT: subcommand

MODEL INDIRECT: offers an alternative to MODEL CONSTRAINT: for models containing indirect effects, where these are not moderated. To use MODEL INDIRECT: instead, you would edit the code above as follows:

First, you can remove the naming of parameters using parentheses in the MODEL: command, i.e. you just need:

```
MODEL:
   Y ON X M1 M2;
   M1 M2 ON X;
```

Second, replace the MODEL CONSTRAINT: subcommand with the following MODEL INDIRECT: subcommand:

```
 MODEL INDIRECT:
   Y IND M1 X;
   Y IND M2 X;
```

or just with

```
 MODEL INDIRECT:
   Y IND X;
```

Leave the OUTPUT: command unchanged.

Model 4c: 1 or more mediators, in parallel if multiple (example uses 1) [BASIC MEDIATION], dichotomous mediator

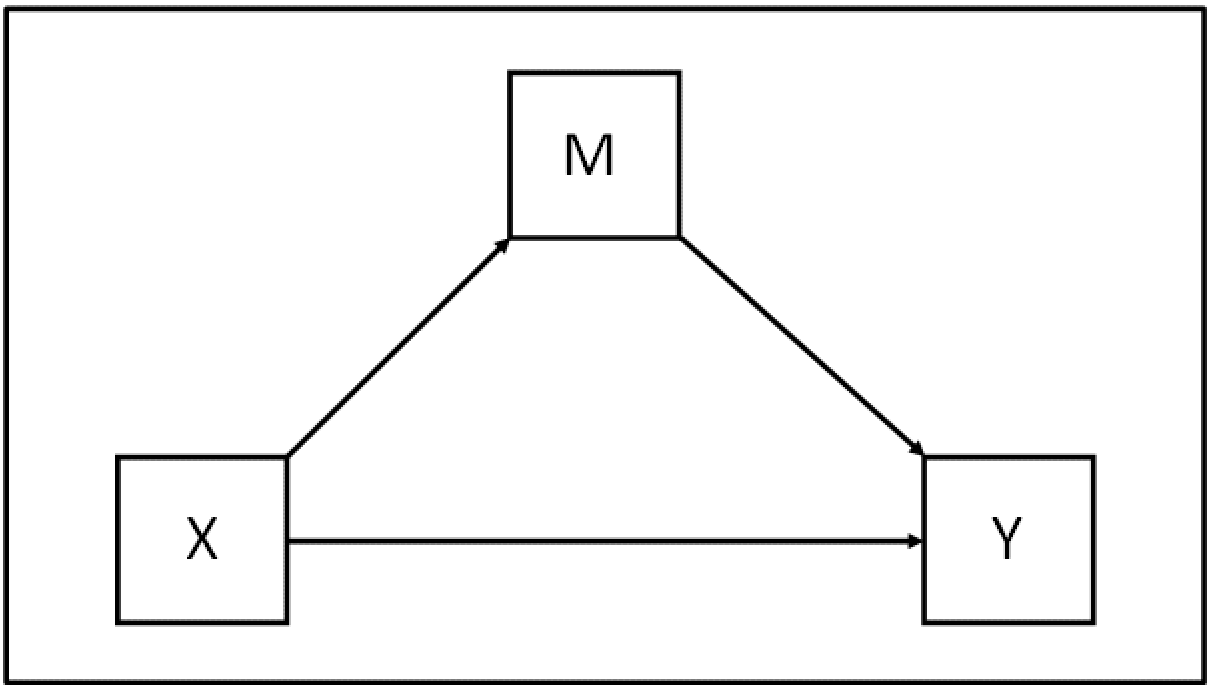
Example Variables: 1 predictor X, 1 mediator M, 1 outcome Y

Preliminary notes:

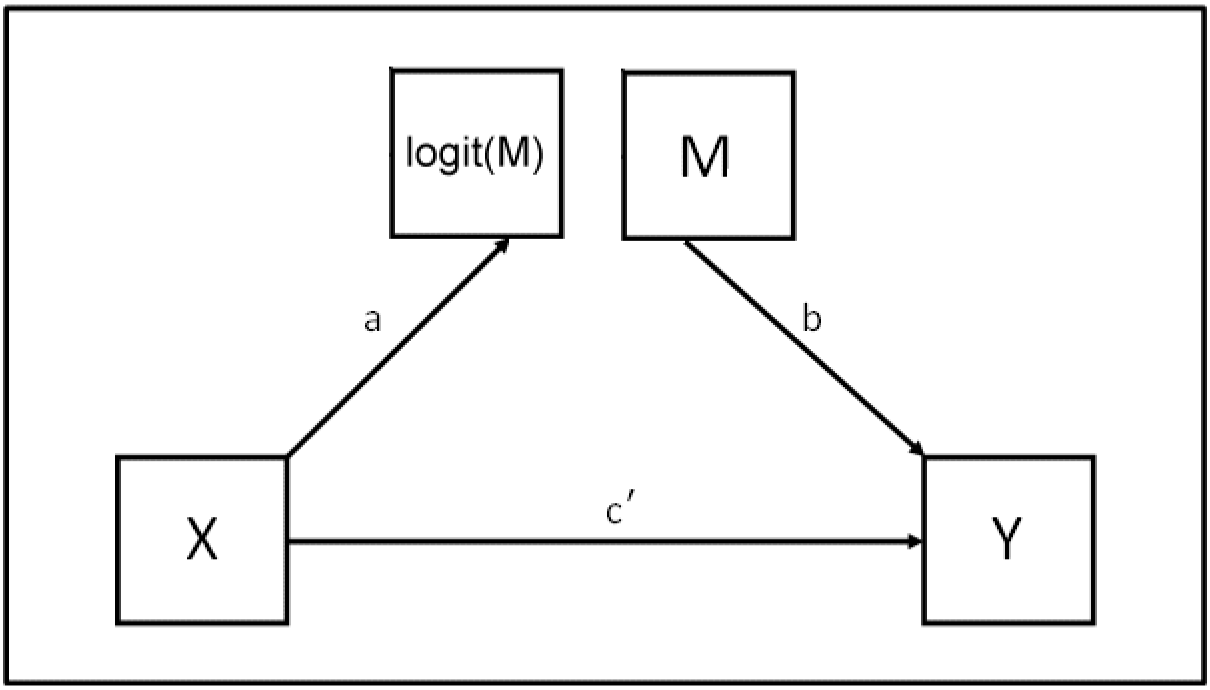
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- The mediator (variable M) is dichotomous.
- The DV (variable Y) is continuous and satisfies the assumptions of standard multiple regression. An example of how to handle a dichotomous DV is given in model 1e (i.e. a moderated logistic regression) and in model 4d (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c'X
logit(M) = a0 + a1X

Algebra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c'X
logit(M) = a0 + a1X

The problem we now have is that we are working on different scales across the two equations. Mplus doesn't automatically acknowledge this (the MODEL INDIRECT command does not yet work in the dichtomous mediator scenario) and adjust for us. So we have to program the method of McKinnon & Dwyer (1993), which adjusts our regression coefficients so that they are on the same scale, to enable the standard decomposition of the total effect into direct and indirect effects i.e. a1*b1 + cdash, to hold. For those who haven't got access to this paper, the maths is kindly outlined by Nathaniel Kerr...

Specifically:

adjusted a1 = a1*SD(X)/(sqrt(a1*a1*Var(X) + pi_squared/3))
adjusted b1 = b1*SD(M)/(sqrt(cdash*cdash*Var(X) + b1*b1*Var(M) + 2*b1*cdash*Cov(X,M) + pi_squared/3))

Hence, when using Mplus, we first need to calculate the sample standard deviations/variances of X and M, and their covariance. This is most easily done by placing these variables into a USEVARIABLES = command, leaving the model section blank, and adding SAMPSTAT; to the output options.

Having calculated those value, you can use the following code...

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - none
! Outcome variable - Y

USEVARIABLES = X M Y;

CATEGORICAL = M;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;

! In model statement name each path using parentheses

MODEL:
   Y ON M (b1);

   Y ON X (cdash);    ! direct effect of X on Y

   M ON X (a1);

! Use model constraint to calculate adjusted indirect effect
! Need to precalculate sample SDs and Covar of X and M, and
input as #'d values below

MODEL CONSTRAINT:
   NEW(SDX SDM COVXM pi adja1 adjb1 adjind);

   SDX = #SDX;
   SDM = #SDM;
   COVXM = #COVXM;
   pi = 3.141592653589793;

   adja1 = a1*SDX/(sqrt((a1*a1*SDX*SDX) + ((pi*pi)/3)));
   adjb1 = b1*SDM/(sqrt((cdash*cdash*SDX*SDX) + (b1*b1*SDM*SDM)
+
     (2*b1*cdash*COVXM) + ((pi*pi)/3)));

   adjind = adja1*adjb1;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 4d: 1 or more mediators, in parallel if multiple (example uses 1) [BASIC MEDIATION], dichotomous outcome

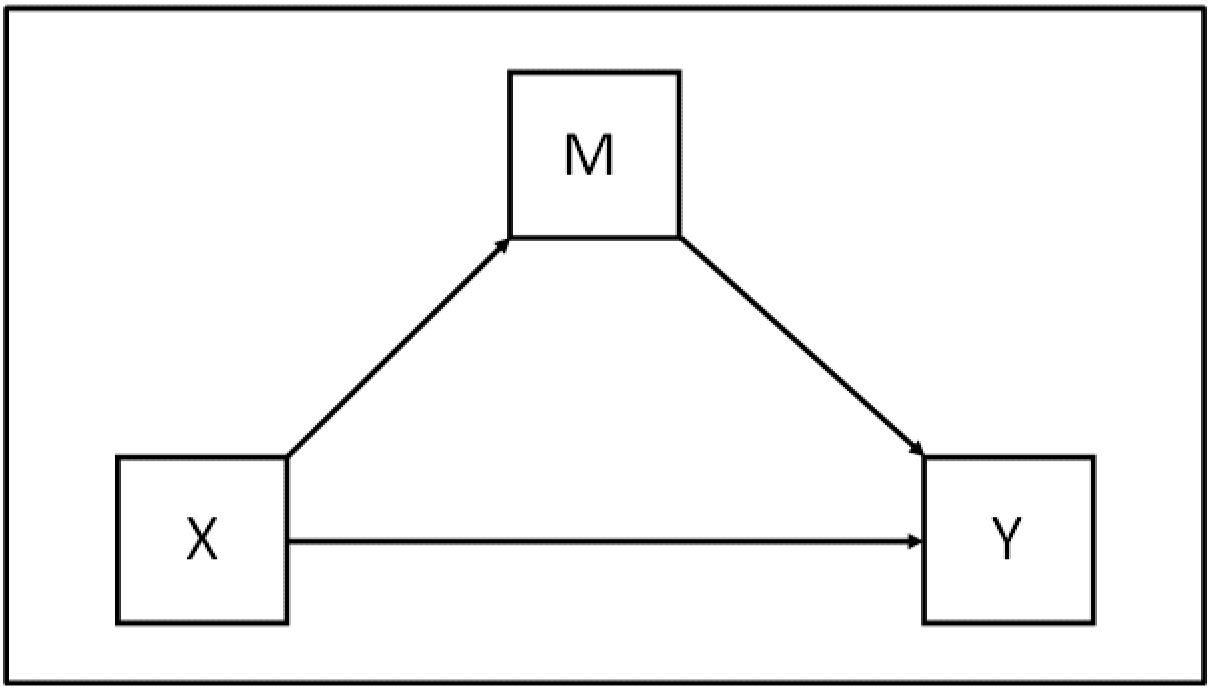
Example Variables: 1 predictor X, 1 mediator M, 1 outcome Y

Preliminary notes:

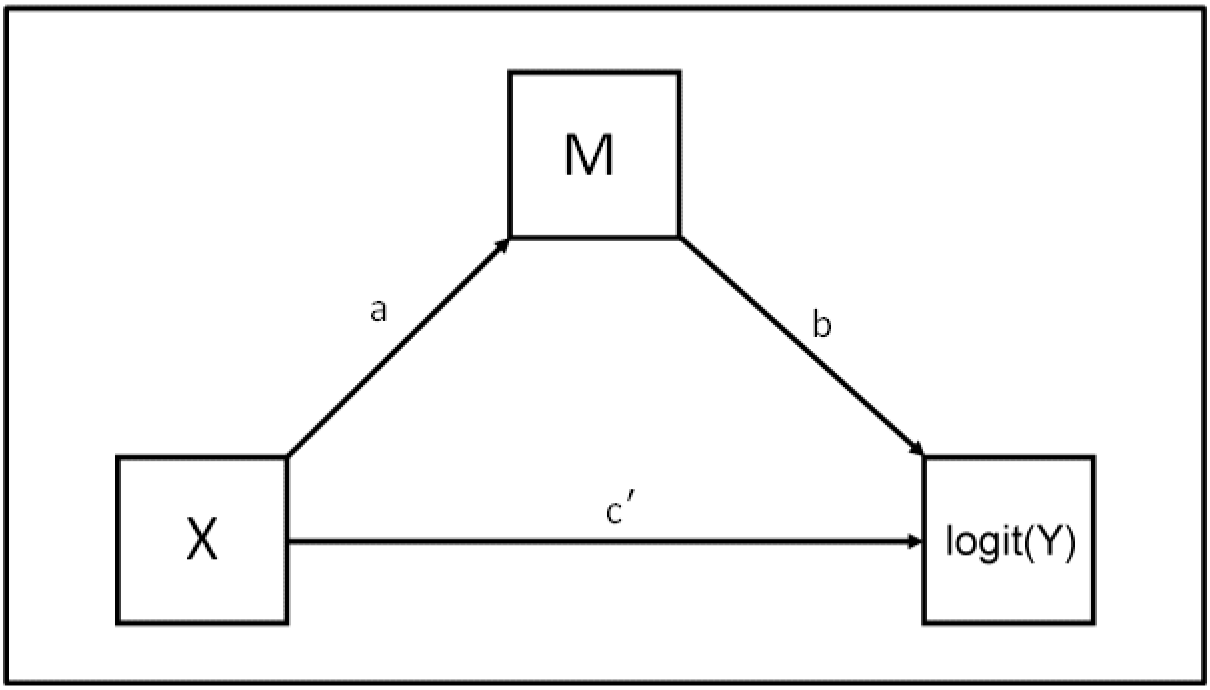
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- The mediator (variable M) is continuous. An example of how to handle a dichotomous mediator is given in model 4c.
- The DV (variable Y) is dichotomous and satisfies the assumptions of binary logistic regression.

Model Diagram:



Statistical Diagram:



Model Equation(s):

logit(Y) = b0 + b1M + c'X
M = a0 + a1X

Algebra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

logit(Y) = b0 + b1M + c'X
M = a0 + a1X

Hence... substituting in equations for M

logit(Y) = b0 + b1(a0 + a1X) + c'X

Hence... multiplying out brackets

logit(Y) = b0 + a0b1 + a1b1X + c'X

Hence... grouping terms into form Y = a + bX

logit(Y) = (b0 + a0b1) + (a1b1 + c')X

Hence...

Indirect effect of X on Y:

a1b1 - or, if expressed as an odds ratio, exp(a1b1)

Direct effect of X on Y:

c' - or, if expressed as an odds ratio, exp(c')

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) - none
! Outcome variable - Y

USEVARIABLES = X M Y;

CATEGORICAL = Y;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;

! In model statement name each path using parentheses

MODEL:
   Y ON M (b1);

   Y ON X (cdash);    ! direct effect of X on Y

   M ON X (a1);

! Use model constraint to calculate indirect effect, and odds
ratio

MODEL CONSTRAINT:
   NEW(a1b1 ORa1b1);
   a1b1 = a1*b1;    ! Indirect effect of X on Y via M
   ORa1b1 = exp(a1*b1);    ! Odds ratio wrto indirect effect of
X on Y via M

OUTPUT:
   STAND CINT(bcbootstrap);
```

Editing required for testing indirect effect(s) using alternative MODEL INDIRECT: subcommand

MODEL INDIRECT: offers an alternative to MODEL CONSTRAINT: for models containing indirect effects, where these are not moderated. To use MODEL INDIRECT: instead, you would edit the code above as follows:

First, you can remove the naming of parameters using parentheses in the MODEL: command, i.e. you just need:

```
MODEL:
   Y ON X M;
   M ON X;
```

Second, replace the MODEL CONSTRAINT: subcommand with the following MODEL INDIRECT: subcommand:

```
   MODEL INDIRECT:
   Y IND X;
```

Leave the OUTPUT: command unchanged.

Model 5: 1 or more mediators, in parallel if multiple, 1 moderator of direct IV-DV path only

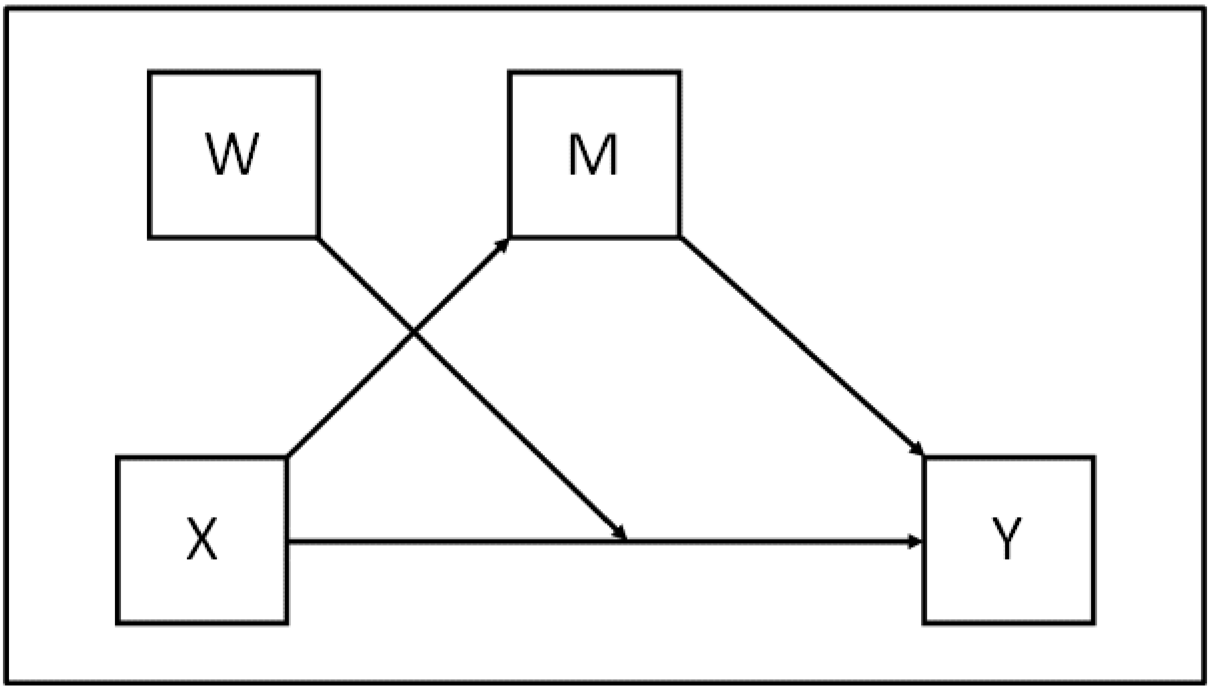
Example Variables: 1 predictor X, 1 mediator M, 1 moderator W, 1 outcome Y

Preliminary notes:

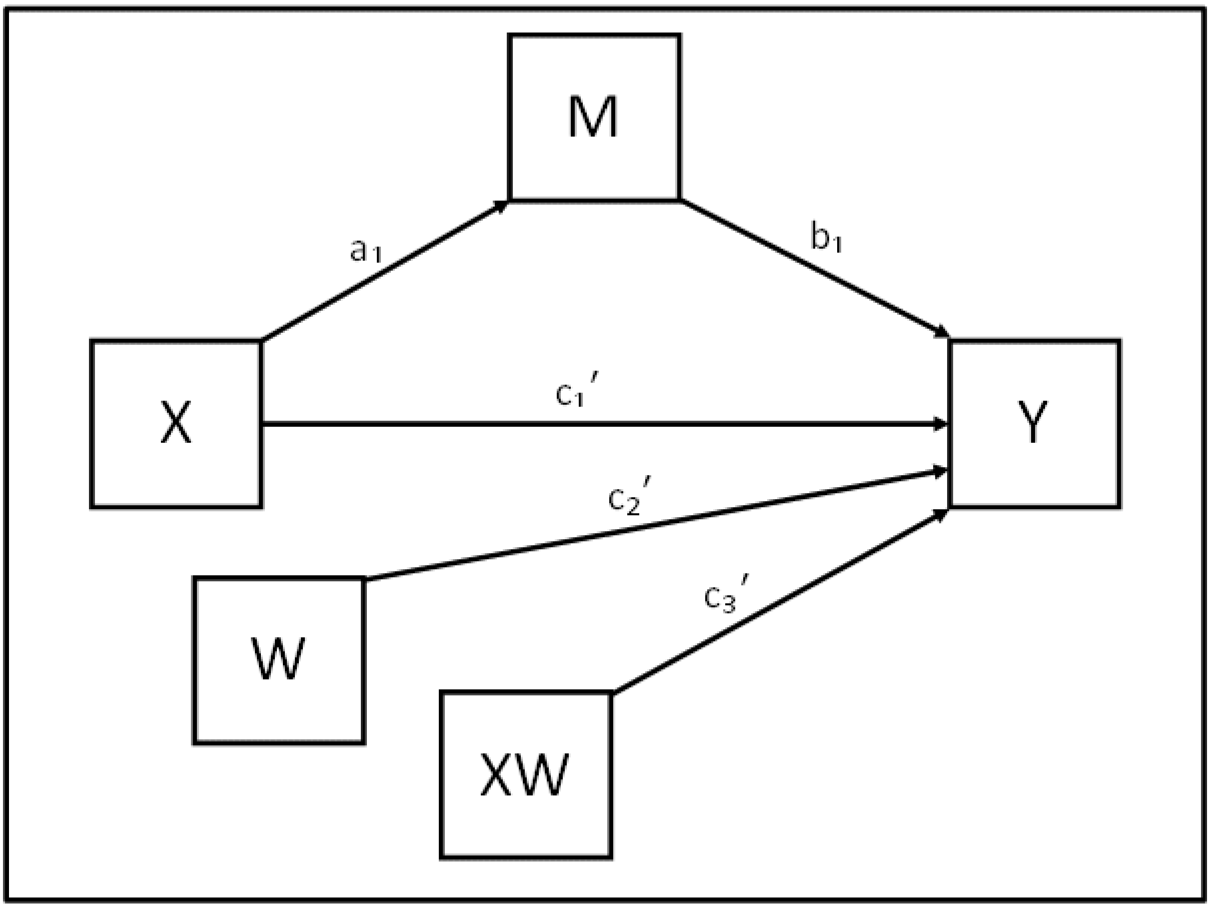
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- Any moderators (variables W,V,Q,Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation)
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c1'X + c2'W + c3'XW
M = a0 + a1X

Albegra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c1'X + c2'W + c3'XW
M = a0 + a1X

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X) + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + c2'W) + (a1b1 + c1' + c3'W)X

Hence...

One indirect effect of X on Y:

a1b1

One direct effect of X on Y, conditional on W:

c1' + c3'W

M*plus* code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W
! Outcome variable - Y

USEVARIABLES = X M W Y XW;

! Create interaction term
! Note that it has to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);

   [M] (a0);
   M ON X (a1);

! Use model constraint subcommand to test simple slopes
! You need to pick low, medium and high moderator values,
```

```
! for example, of 1 SD below mean, mean, 1 SD above mean
! Also calc total effects at lo, med, hi values of moderator

MODEL CONSTRAINT:
   NEW(LOW_W MED_W HIGH_W a1b1 DIR_LO DIR_MED DIR_HI TOT_LO
TOT_MED TOT_HI);

   LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
   MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
   HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

! Now calc indirect effect - and conditional direct effects
for each value of W

   a1b1 = a1*b1;

   DIR_LO = cdash1 + cdash3*LOW_W;
   DIR_MED = cdash1 + cdash3*MED_W;
   DIR_HI = cdash1 + cdash3*HIGH_W;

   TOT_LO = DIR_LO + a1b1;
   TOT_MED = DIR_MED + a1b1;
   TOT_HI = DIR_HI + a1b1;

! Use loop plot to plot total effect of X on Y for low, med,
high values of W
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

   PLOT(LOMOD MEDMOD HIMOD);

   LOOP(XVAL,1,5,0.1);

   LOMOD = (b0 + a0*b1 + cdash2*LOW_W) + TOT_LO*XVAL;
   MEDMOD = (b0 + a0*b1 + cdash2*MED_W) + TOT_MED*XVAL;
   HIMOD = (b0 + a0*b1 + cdash2*HIGH_W) + TOT_HI*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 6: 2 or more mediators (2 in this example), in series

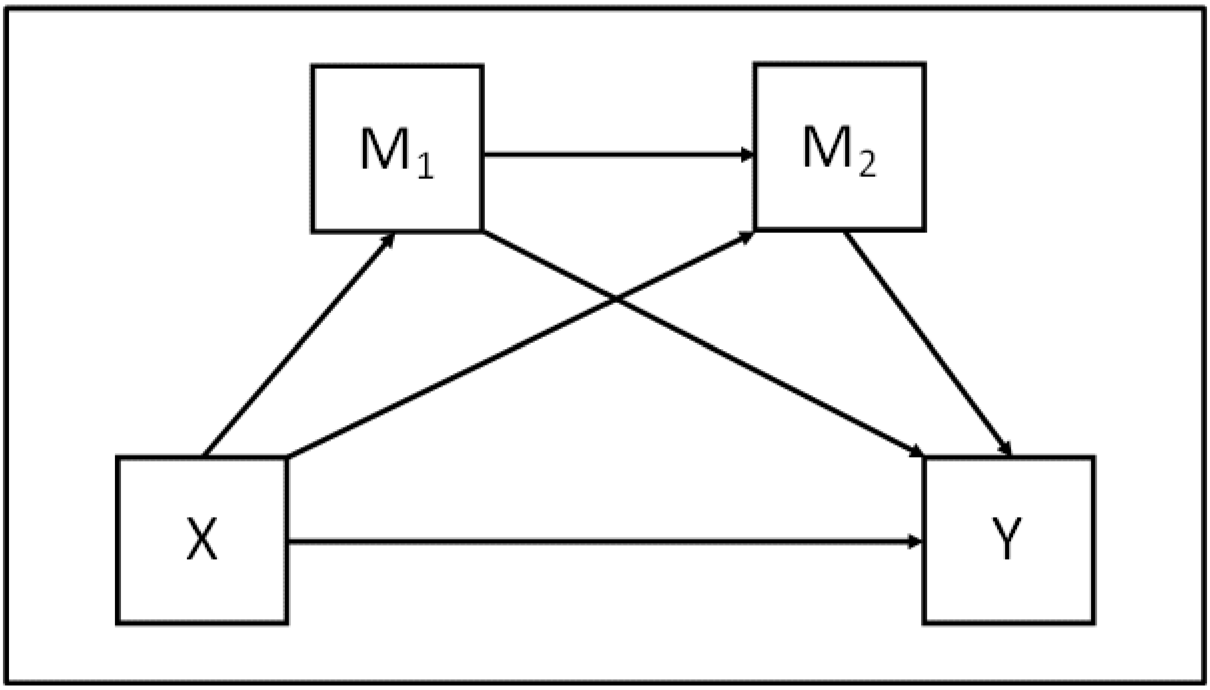
Example Variables: 1 predictor X, 2 mediators M1 and M2, 1 outcome Y

Preliminary notes:

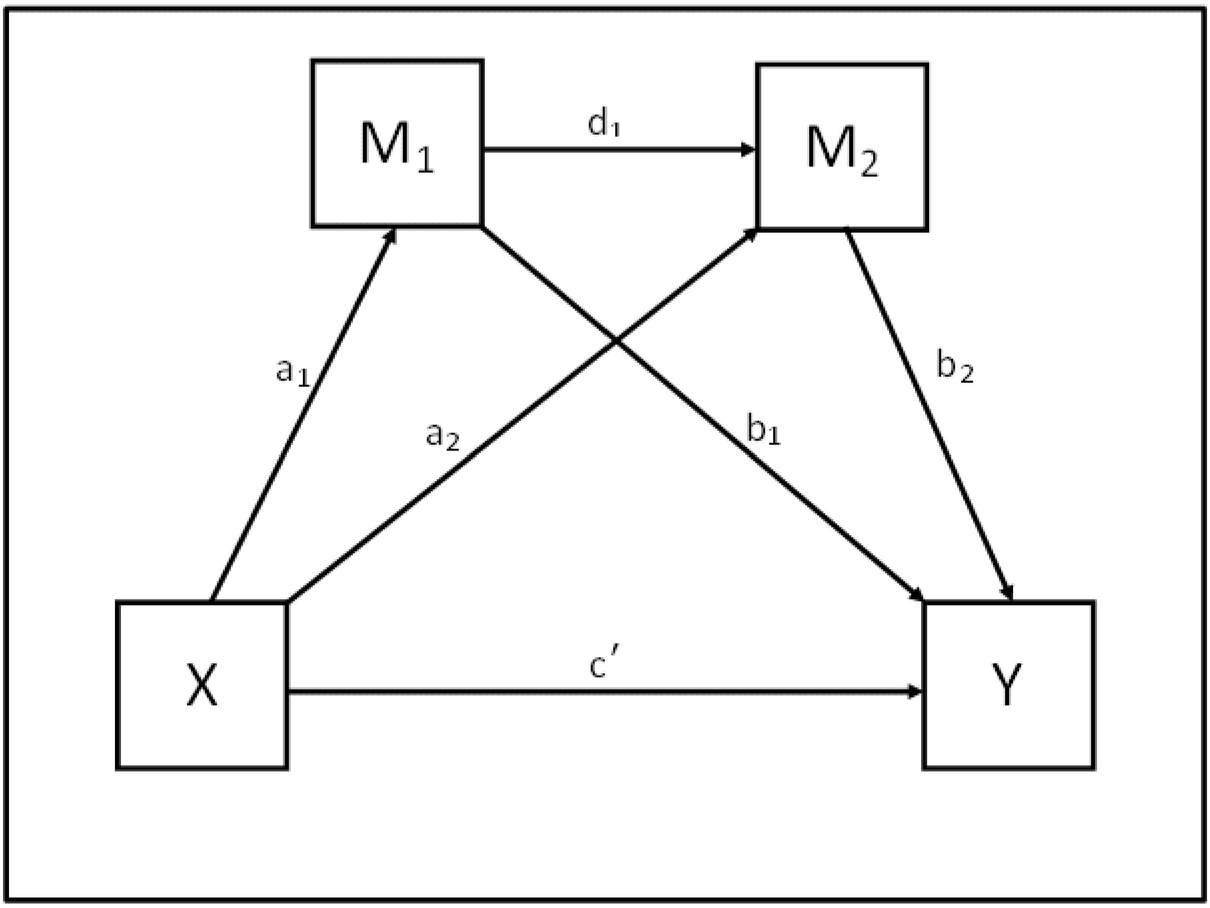
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- Any moderators (variables W,V,Q,Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation)
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M1 + b2M2 + c'X
M1 = a01 + a1X
M2 = a02 + a2X + d1M1

Albegra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M1 + b2M2 + c'X
M1 = a01 + a1X
M2 = a02 + a2X + d1M1

Hence... substituting in equations for M1 and M2

Y = b0 + b1(a01 + a1X) + b2(a02 + a2X + d1(a01 + a1X)) + c'X

Hence... multiplying out brackets

Y = b0 + a01b1 + a1b1X + a02b2 + a2b2X + a01d1b2 + a1d1b2X + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a01b1 + a02b2 + a01d1b2) + (a1b1 + a2b2 + a1d1b2 + c')X

Hence... Three indirect effects of X on Y: a1b1, a2b2, a1d1b2

One direct effect of X on Y: c'

M*plus* code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M1, M2
! Moderator variable(s) - none
! Outcome variable - Y

USEVARIABLES = X M1 M2 Y;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path using parentheses

MODEL:
   Y ON X (cdash);    ! direct effect of X on Y
   Y ON M1 (b1);
   Y ON M2 (b2);

   M1 ON X (a1);
   M2 ON X (a2);
   M2 ON M1 (d1);

! Use model constraint to calculate specific indirect paths
and total indirect effect

MODEL CONSTRAINT:
   NEW(a1b1 a2b2 a1d1b2 TOTALIND TOTAL);
   a1b1 = a1*b1;    ! Specific indirect effect of X on Y via
M1
   a2b2 = a2*b2;    ! Specific indirect effect of X on Y via
M2
   a1d1b2 = a1*d1*b2;    ! Specific indirect effect of X on Y
via M1 and M2
   TOTALIND = a1*b1 + a2*b2 + a1*d1*b2;    ! Total indirect
effect of X on Y via M1, M2
   TOTAL = a1*b1 + a2*b2 + a1*d1*b2 + cdash;    ! Total effect
of X on Y

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 7: 1 or more mediators, in parallel if multiple (example uses 1), 1 moderator of IV-Mediator path only

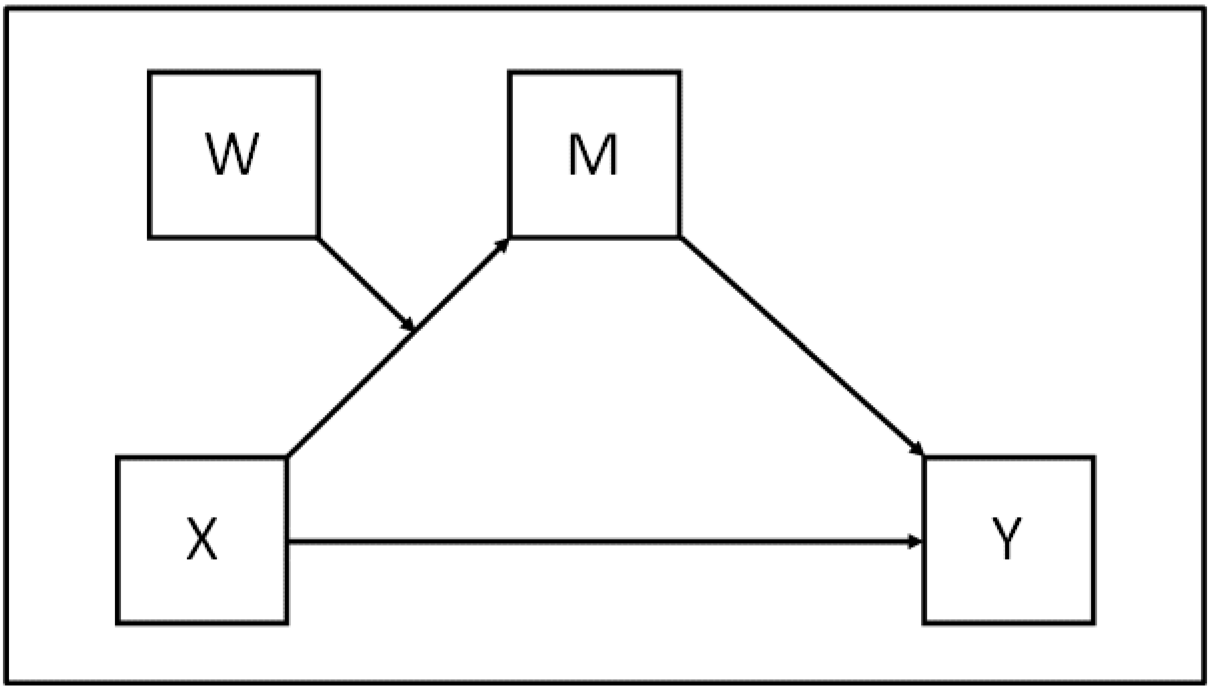
Example Variables: 1 predictor X, 1 mediator M, 1 moderator W, 1 outcome Y

Preliminary notes:

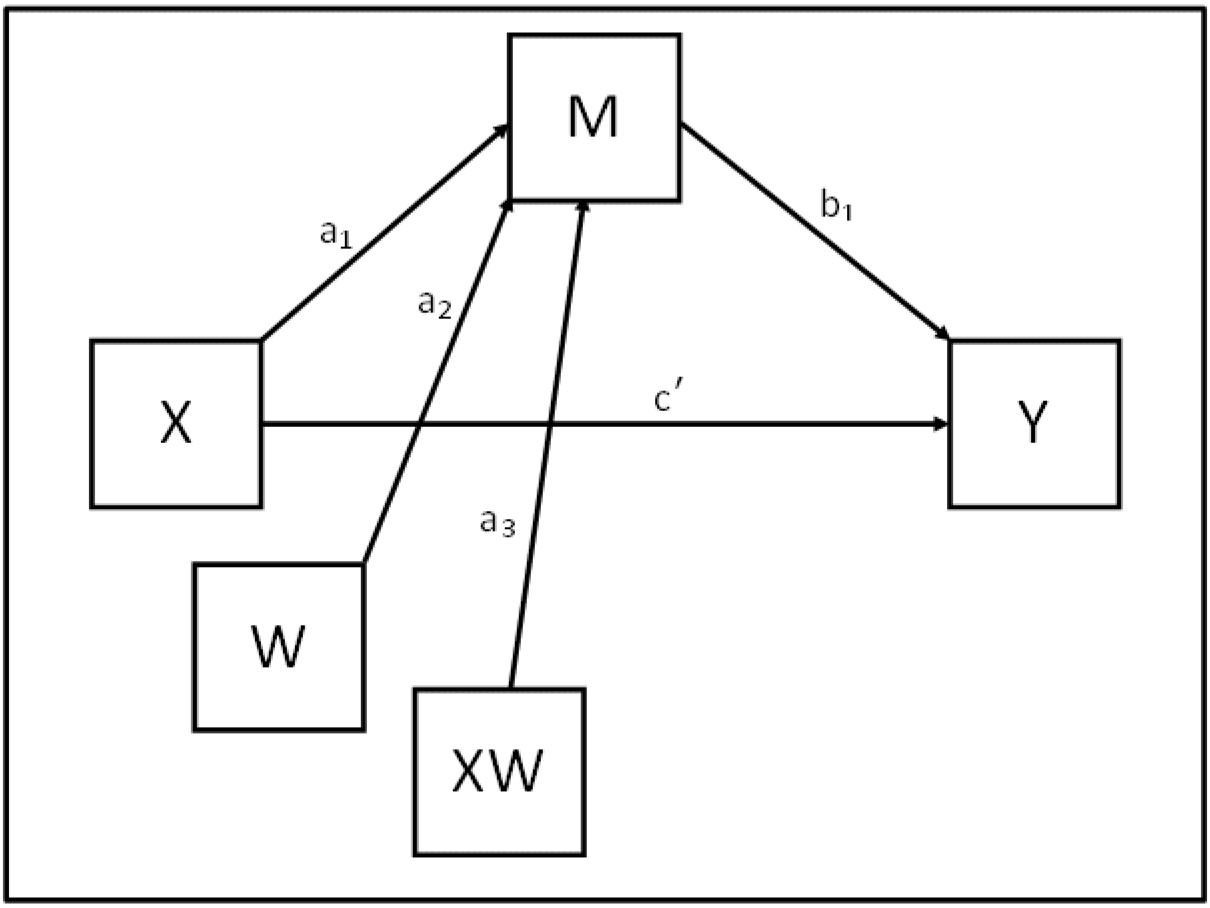
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c'X
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c'X
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W) + (a1b1 + a3b1W + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W: a1b1 + a3b1W = (a1 + a3W)b1

One direct effect of X on Y: c'

M*plus* code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W
! Outcome variable - Y

USEVARIABLES = X M W Y XW;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);

   Y ON X (cdash);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for
W
! for example, of 1 SD below mean, mean, 1 SD above mean
```

```
! 1 moderator, 3 values for it
! arbitrary naming convention for conditional indirect and
total effects used below:
! MED_Q = medium value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W
    IND_LOWW IND_MEDW IND_HIW
    TOT_LOWW TOT_MEDW TOT_HIW);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

! Calc conditional indirect effects for each combination of
moderator values

    IND_LOWW = a1*b1 + a3*b1*LOW_W;
    IND_MEDW = a1*b1 + a3*b1*MED_W;
    IND_HIW = a1*b1 + a3*b1*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TOT_LOWW = IND_LOWW + cdash;
    TOT_MEDW = IND_MEDW + cdash;
    TOT_HIW = IND_HIW + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(LOMOD MEDMOD HIMOD);

    LOOP(XVAL,1,5,0.1);

    LOMOD = IND_LOWW*XVAL;
    MEDMOD = IND_MEDW*XVAL;
    HIMOD = IND_HIW*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 8: 1 or more mediators, in parallel if multiple (example uses 1), 1 moderator moderating both the IV-Mediator path and direct IV-DV path

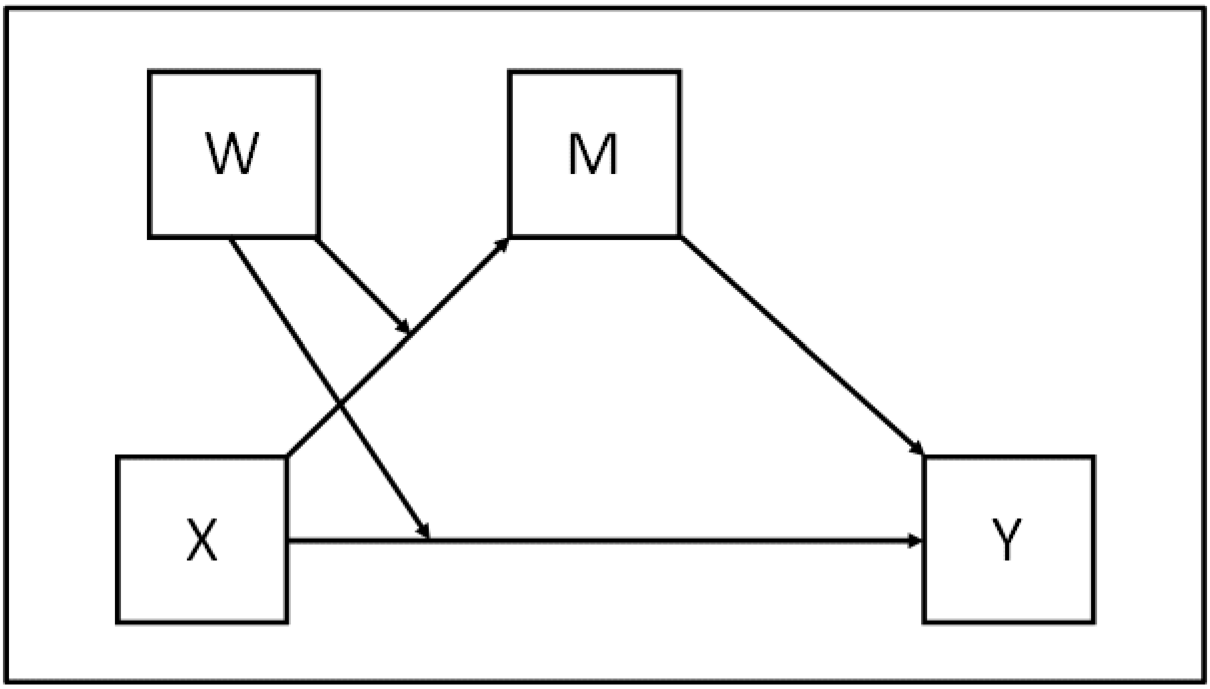
Example Variables: 1 predictor X, 1 mediator M, 1 moderator W, 1 outcome Y

Preliminary notes:

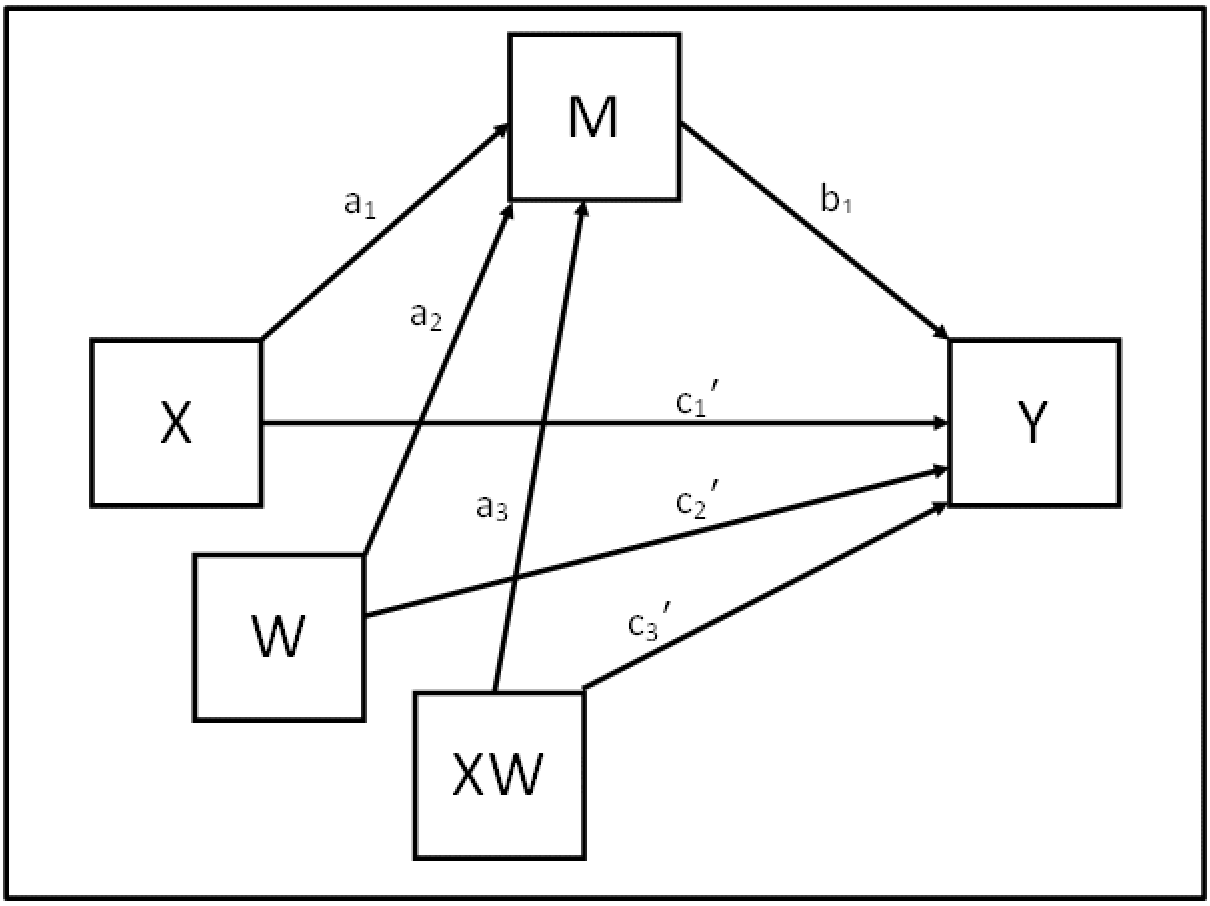
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX
Y = (b0 + a0b1 + a2b1W + c2'W) + (a1b1 + a3b1W + c1' + c3'W)X
Hence...One indirect effect(s) of X on Y, conditional on W: a1b1 + a3b1W = (a1 + a3W)b1
One direct effect of X on Y, conditional on W: c1' + c3'W

M*plus* code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W
! Outcome variable - Y

USEVARIABLES = X M W Y XW;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for
W
! for example, of 1 SD below mean, mean, 1 SD above mean

! 1 moderator, 3 values for it
! arbitrary naming convention for conditional indirect and
total effects used below:
! MED_Q = medium value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W
    IND_LOWW IND_MEDW IND_HIW
```

```
    DIR_LOWW DIR_MEDW DIR_HIW
    TOT_LOWW TOT_MEDW TOT_HIW);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

! Calc conditional indirect effects for each combination of
moderator values

    IND_LOWW = a1*b1 + a3*b1*LOW_W;
    IND_MEDW = a1*b1 + a3*b1*MED_W;
    IND_HIW = a1*b1 + a3*b1*HIGH_W;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWW = cdash1 + cdash3*LOW_W;
    DIR_MEDW = cdash1 + cdash3*MED_W;
    DIR_HIW = cdash1 + cdash3*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TOT_LOWW = IND_LOWW + DIR_LOWW;
    TOT_MEDW = IND_MEDW + DIR_MEDW;
    TOT_HIW = IND_HIW + DIR_HIW;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(LOMOD MEDMOD HIMOD);

    LOOP(XVAL,1,5,0.1);

    LOMOD = IND_LOWW*XVAL;
    MEDMOD = IND_MEDW*XVAL;
    HIMOD = IND_HIW*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 9: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the IV-Mediator path only

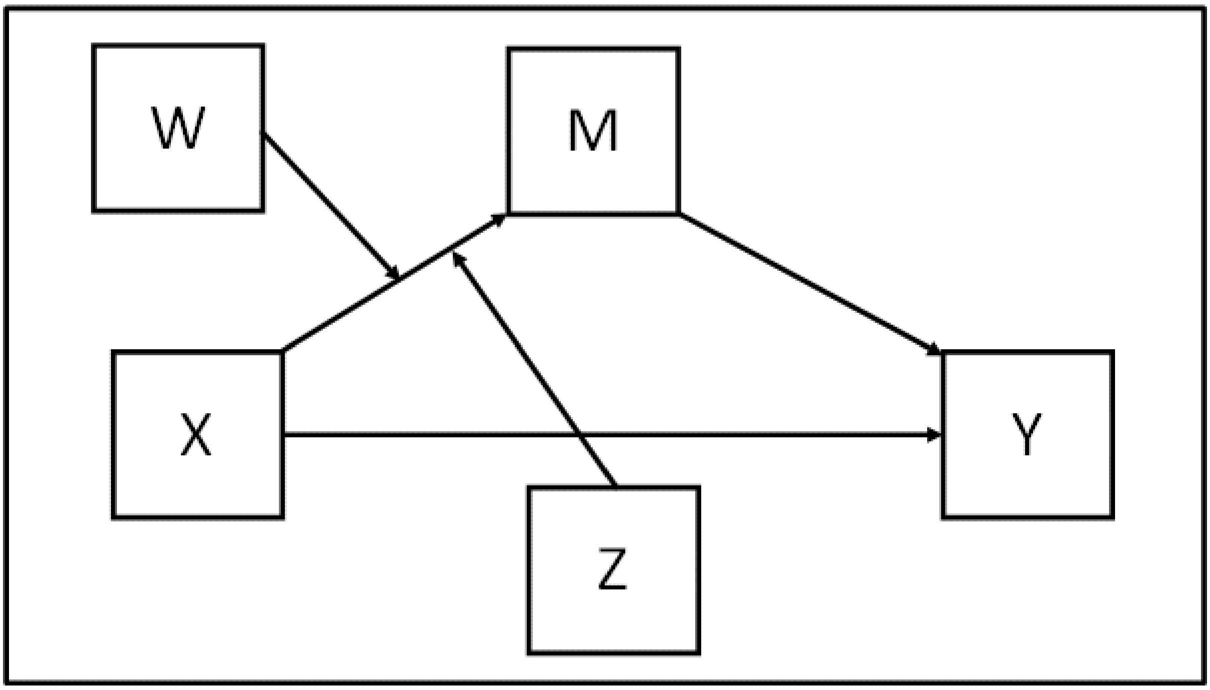
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

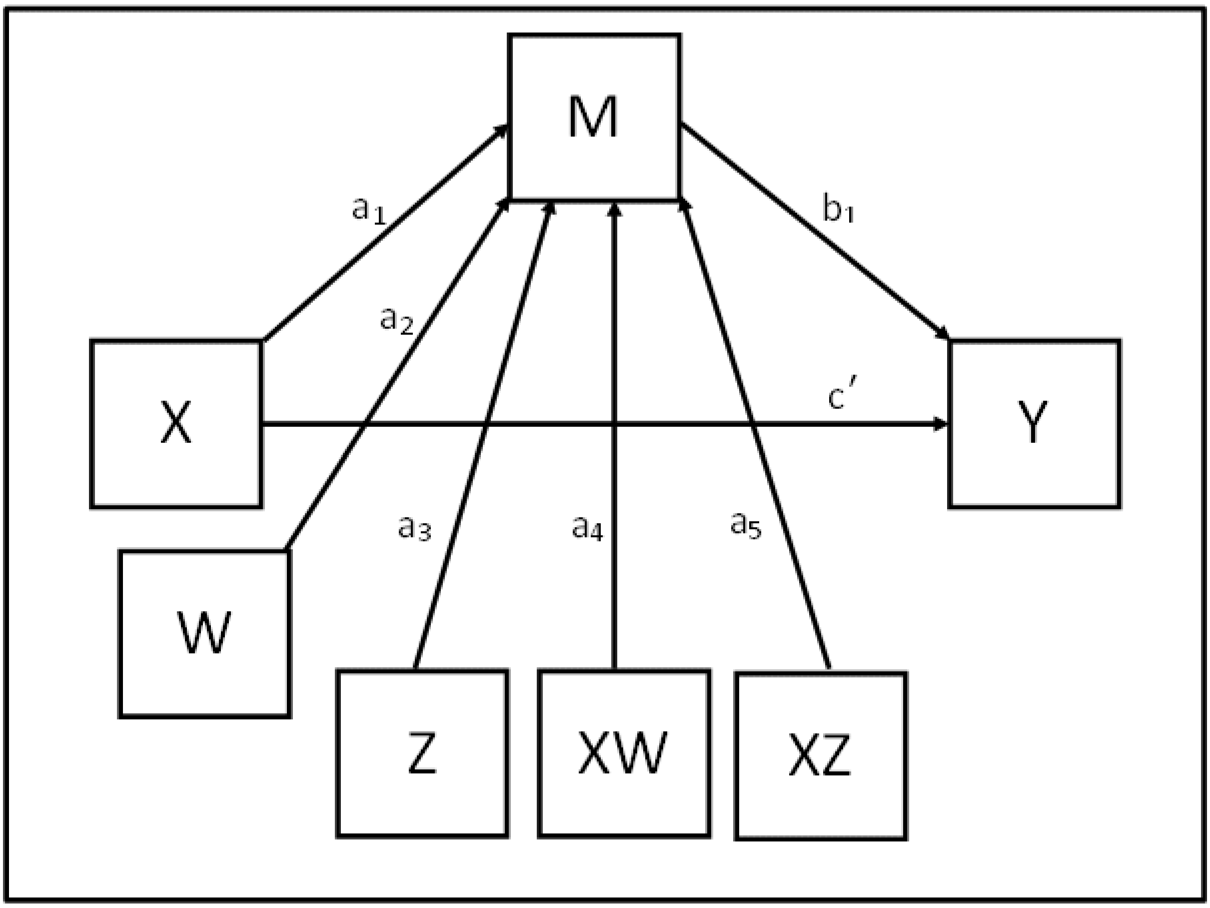
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z) + (a1b1 + a4b1W + a5b1Z + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z = (a1 + a4W + a5Z)b1

One direct effect of X on Y:

c'

M*plus* code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);

   Y ON X (cdash);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
```

```
! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values
```

```
    TLOW_LOZ = ILOW_LOZ + cdash;
    TMEW_LOZ = IMEW_LOZ + cdash;
    THIW_LOZ = IHIW_LOZ + cdash;

    TLOW_MEZ = ILOW_MEZ + cdash;
    TMEW_MEZ = IMEW_MEZ + cdash;
    THIW_MEZ = IHIW_MEZ + cdash;

    TLOW_HIZ = ILOW_HIZ + cdash;
    TMEW_HIZ = IMEW_HIZ + cdash;
    THIW_HIZ = IHIW_HIZ + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 10: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating both the IV- Mediator path and direct IV-DV path

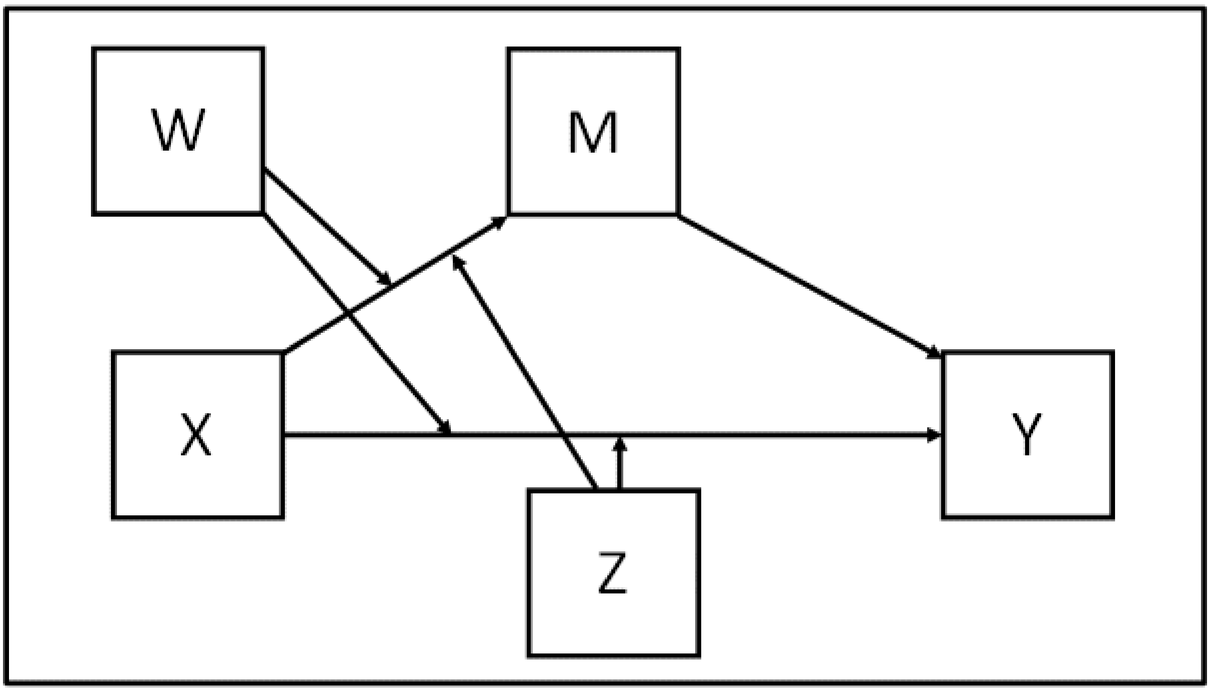
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

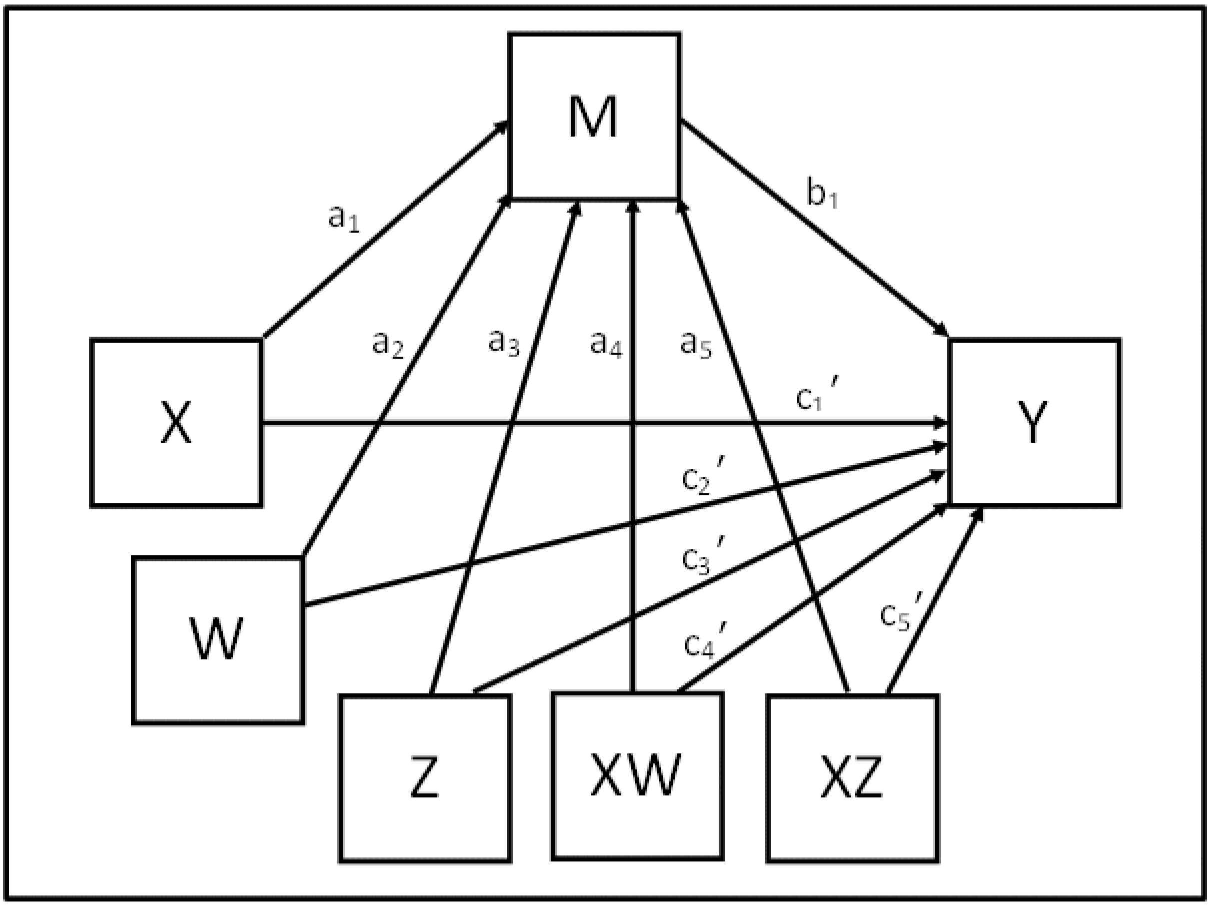
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M
Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... multiplying out brackets
Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + c2'W + c3'Z) + (a1b1 + a4b1W + a5b1Z + c1' + c4'W + c5'Z)X

Hence... One indirect effect(s) of X on Y, conditional on W, Z: a1b1 + a4b1W + a5b1Z = (a1 + a4W + a5Z)b1

One direct effect of X on Y, conditional on W, Z: c1' + c4'W + c5'Z

M*plus* code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON Z (cdash3);
   Y ON XW (cdash4);
   Y ON XZ (cdash5);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and total
effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.
```

```
MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your chosen
low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your chosen
medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your chosen
low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your chosen
medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z;
```

```
! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + DLOW_LOZ;
    TMEW_LOZ = IMEW_LOZ + DMEW_LOZ;
    THIW_LOZ = IHIW_LOZ + DHIW_LOZ;

    TLOW_MEZ = ILOW_MEZ + DLOW_MEZ;
    TMEW_MEZ = IMEW_MEZ + DMEW_MEZ;
    THIW_MEZ = IHIW_MEZ + DHIW_MEZ;

    TLOW_HIZ = ILOW_HIZ + DLOW_HIZ;
    TMEW_HIZ = IMEW_HIZ + DMEW_HIZ;
    THIW_HIZ = IHIW_HIZ + DHIW_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y for
each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional total
effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 11: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the IV-Mediator path only, all 2-way and 3-way interactions

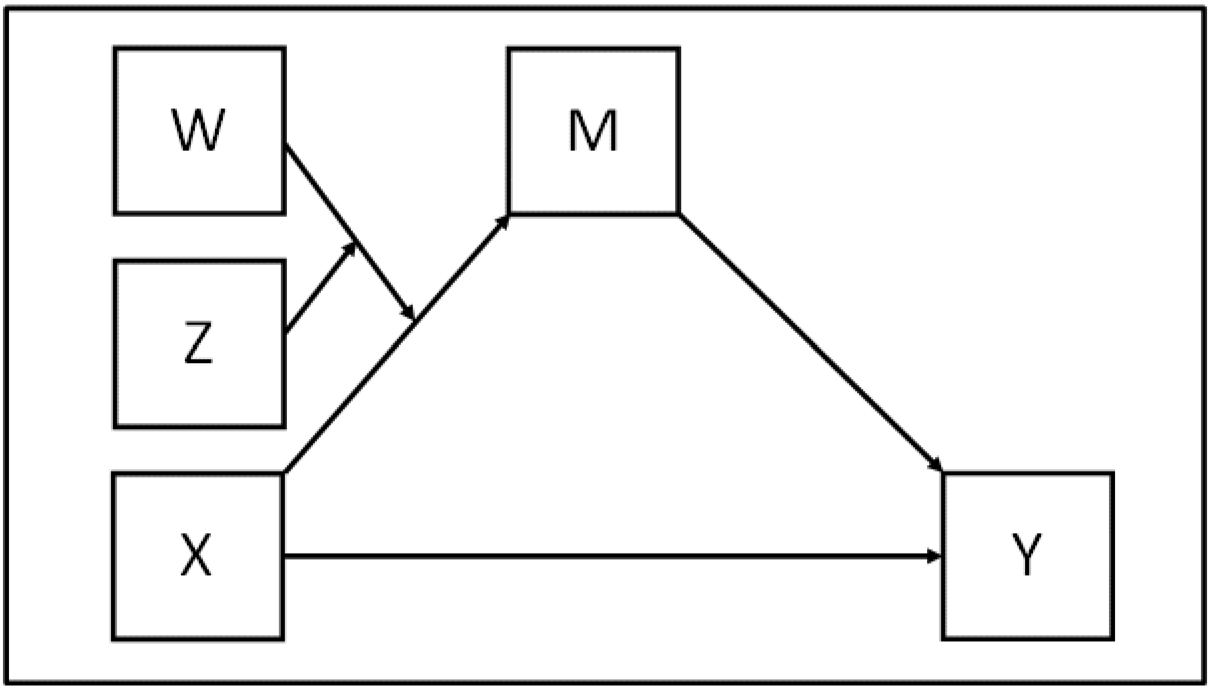
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

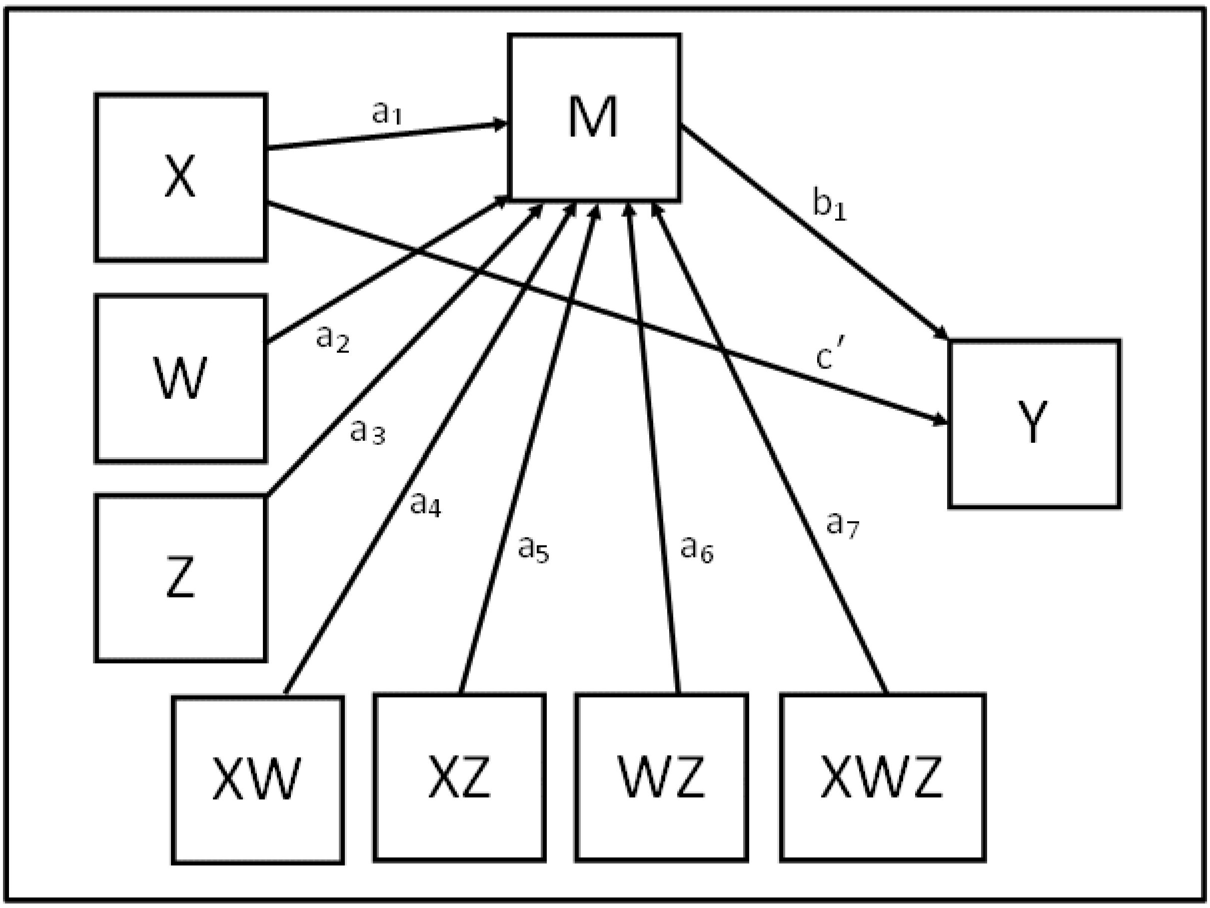
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z: a1b1 + a4b1W + a5b1Z + a7b1WZ = (a1 + a4W + a5Z + a7WZ)b1

One direct effect of X on Y: c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ WZ XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);

   Y ON X (cdash);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
```

```
    M ON WZ (a6);
    M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and total
effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your chosen
low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your chosen
medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your chosen
low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your chosen
medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z;
```

```
    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + cdash;
    TMEW_LOZ = IMEW_LOZ + cdash;
    THIW_LOZ = IHIW_LOZ + cdash;

    TLOW_MEZ = ILOW_MEZ + cdash;
    TMEW_MEZ = IMEW_MEZ + cdash;
    THIW_MEZ = IHIW_MEZ + cdash;

    TLOW_HIZ = ILOW_HIZ + cdash;
    TMEW_HIZ = IMEW_HIZ + cdash;
    THIW_HIZ = IHIW_HIZ + cdash;

! Use loop plot to plot conditional indirect effect of X on Y for
each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional total
effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 12: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the IV-Mediator path and direct IV-DV path, all 2-way and 3-way interactions

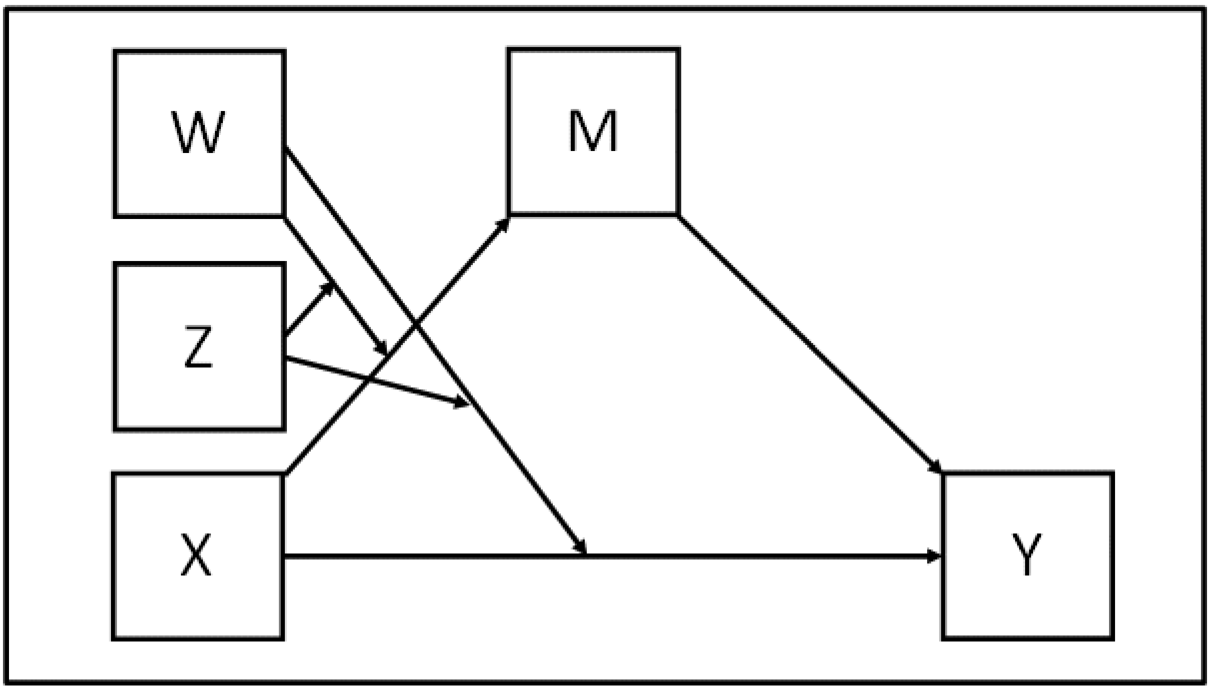
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

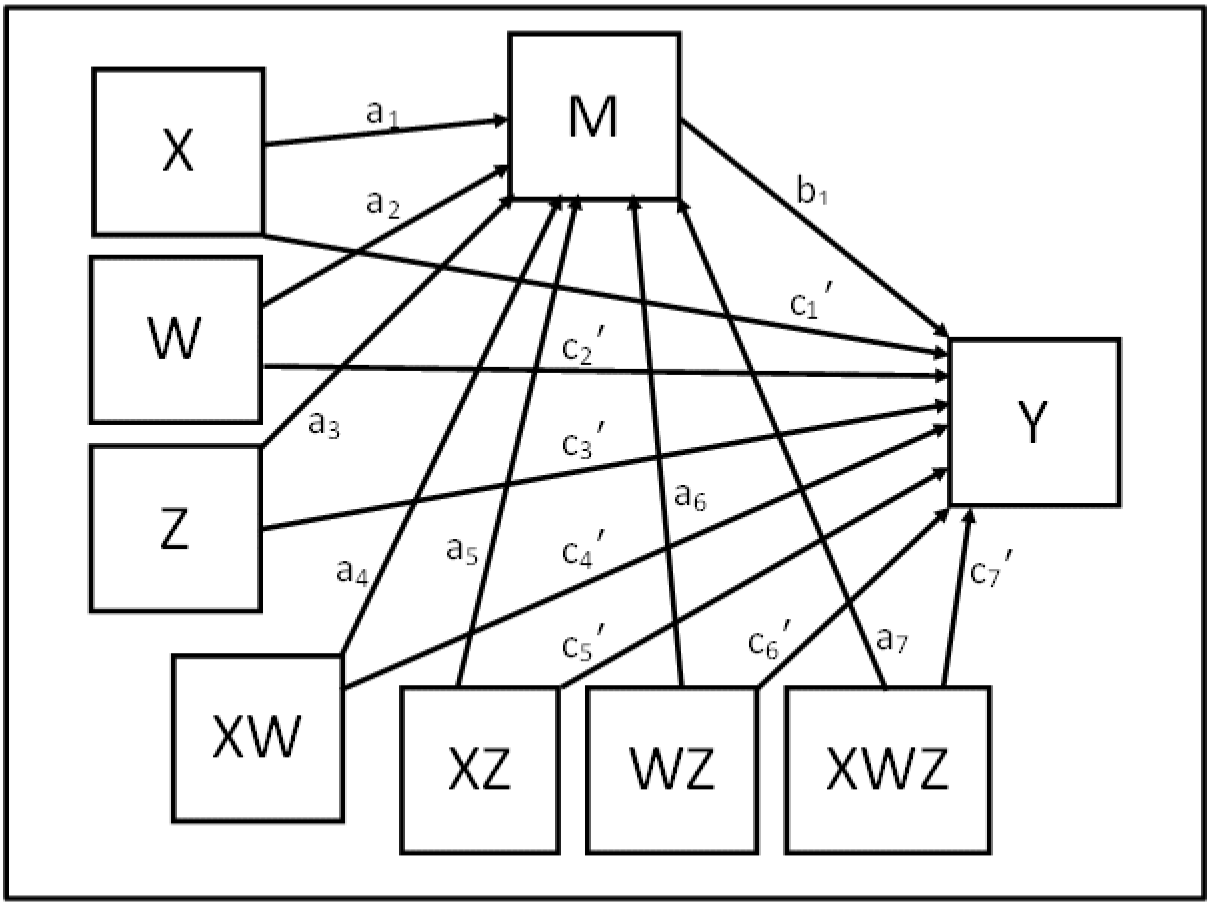
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + c2'W + c3'Z + c6'WZ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + c1' + c4'W + c5'Z + c7'WZ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a7b1WZ = (a1 + a4W + a5Z + a7WZ)b1

One direct effect of X on Y, conditional on W, Z:

c1' + c4'W + c5'Z + c7'WZ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ WZ XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON Z (cdash3);
   Y ON XW (cdash4);
```

```
    Y ON XZ (cdash5);
    Y ON WZ (cdash6);
    Y ON XWZ (cdash7);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
    M ON WZ (a6);
    M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values
```

```
    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + DLOW_LOZ;
    TMEW_LOZ = IMEW_LOZ + DMEW_LOZ;
    THIW_LOZ = IHIW_LOZ + DHIW_LOZ;
```

```
    TLOW_MEZ = ILOW_MEZ + DLOW_MEZ;
    TMEW_MEZ = IMEW_MEZ + DMEW_MEZ;
    THIW_MEZ = IHIW_MEZ + DHIW_MEZ;

    TLOW_HIZ = ILOW_HIZ + DLOW_HIZ;
    TMEW_HIZ = IMEW_HIZ + DMEW_HIZ;
    THIW_HIZ = IHIW_HIZ + DHIW_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 13: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating the IV- Mediator path, 3-way interaction, 1 also moderating direct IV-DV path

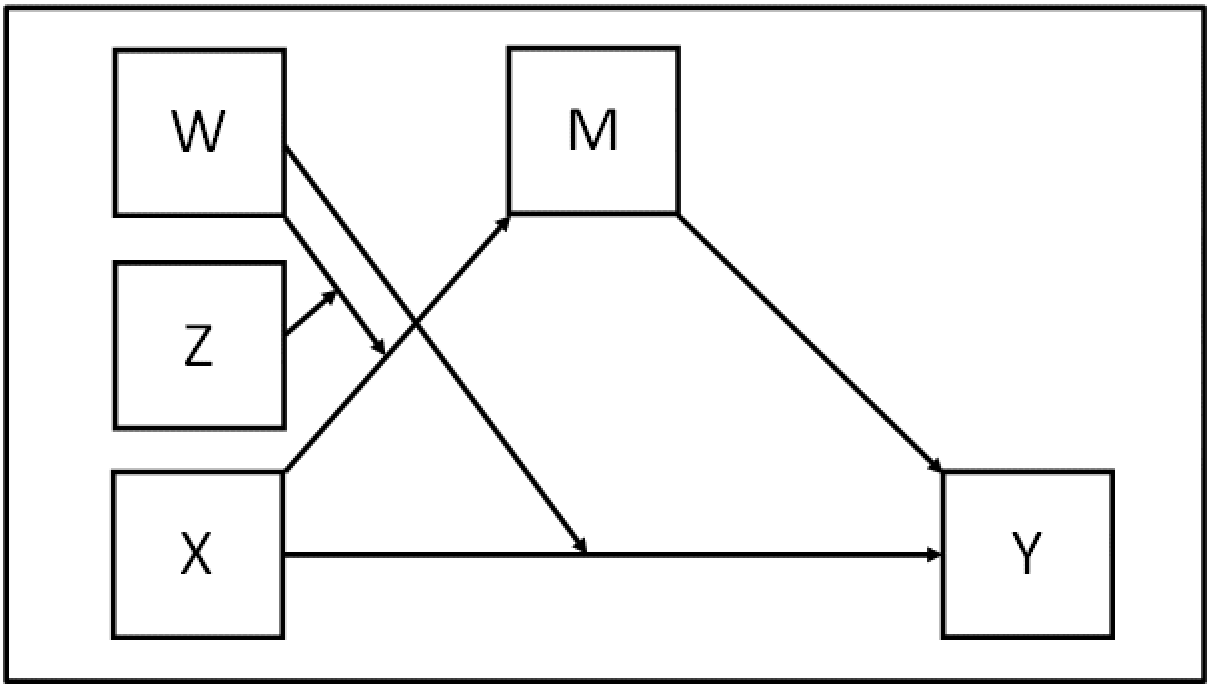
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

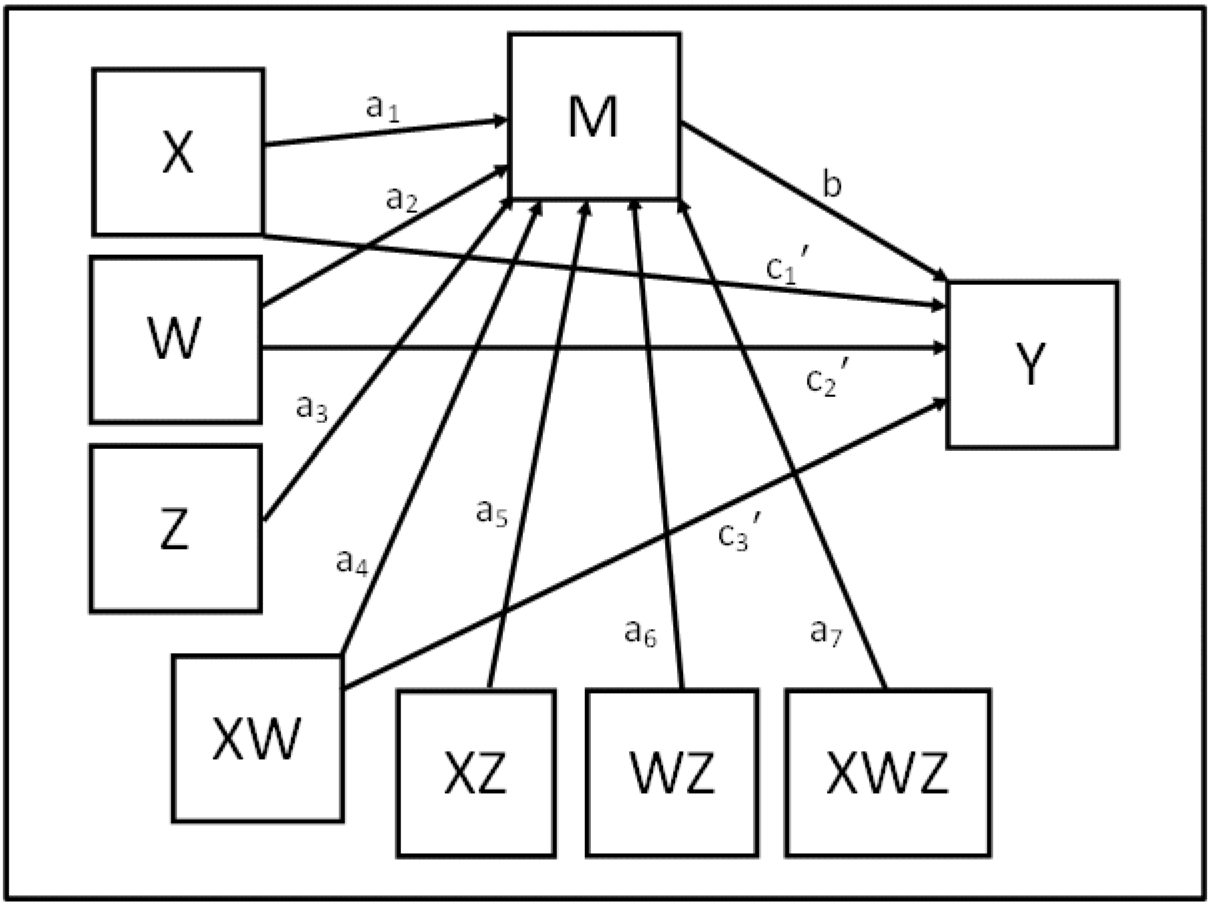
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + c2'W) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + c1' + c3'W)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a7b1WZ = (a1 + a4W + a5Z + a7WZ)b1

One direct effect of X on Y, conditional on W:

c1' + c3'W

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ WZ XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);
```

```
    [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    DIR_LOWW DIR_MEDW DIR_HIW
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z;
```

```
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWW = cdash1 + cdash3*LOW_W;
    DIR_MEDW = cdash1 + cdash3*MED_W;
    DIR_HIW = cdash1 + cdash3*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + DIR_LOWW;
    TMEW_LOZ = IMEW_LOZ + DIR_MEDW;
    THIW_LOZ = IHIW_LOZ + DIR_HIW;

    TLOW_MEZ = ILOW_MEZ + DIR_LOWW;
    TMEW_MEZ = IMEW_MEZ + DIR_MEDW;
    THIW_MEZ = IHIW_MEZ + DIR_HIW;

    TLOW_HIZ = ILOW_HIZ + DIR_LOWW;
    TMEW_HIZ = IMEW_HIZ + DIR_MEDW;
    THIW_HIZ = IHIW_HIZ + DIR_HIW;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);
```

```
     PLOW_LOZ = ILOW_LOZ*XVAL;
     PMEW_LOZ = IMEW_LOZ*XVAL;
     PHIW_LOZ = IHIW_LOZ*XVAL;

     PLOW_MEZ = ILOW_MEZ*XVAL;
     PMEW_MEZ = IMEW_MEZ*XVAL;
     PHIW_MEZ = IHIW_MEZ*XVAL;

     PLOW_HIZ = ILOW_HIZ*XVAL;
     PMEW_HIZ = IMEW_HIZ*XVAL;
     PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 14: 1 or more mediators, in parallel if multiple (example uses 1), 1 moderator of Mediator-DV path only

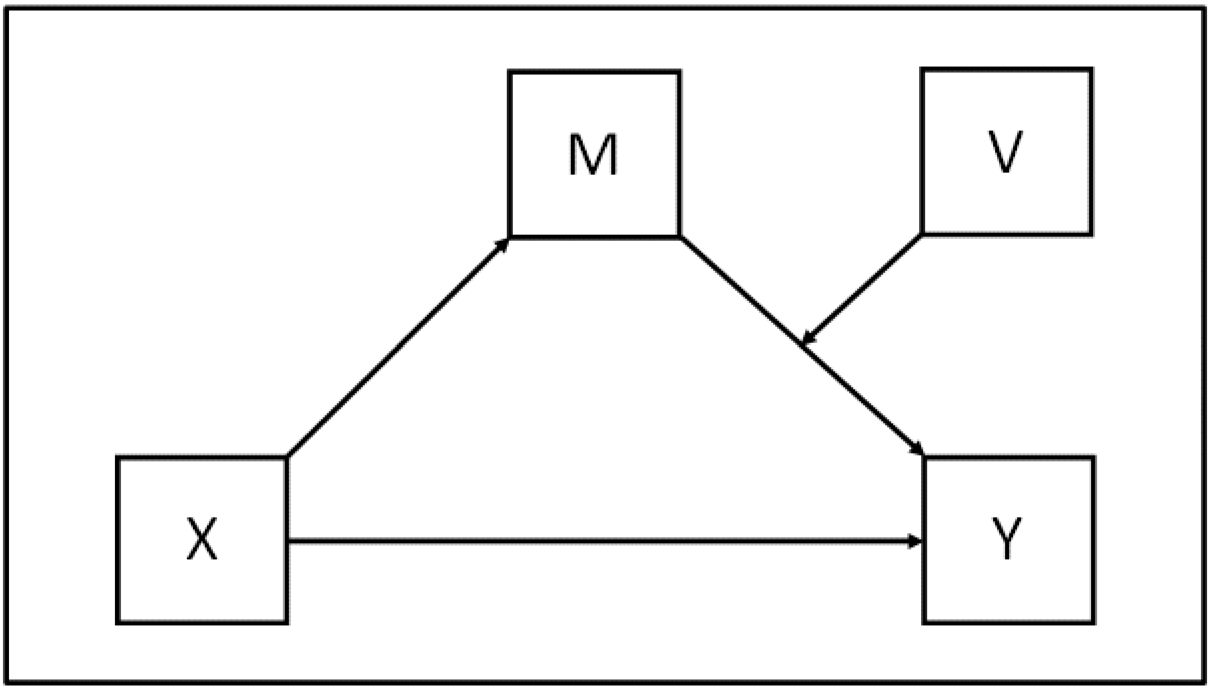
Example Variables: 1 predictor X, 1 mediator M, 1 moderator V, 1 outcome Y

Preliminary notes:

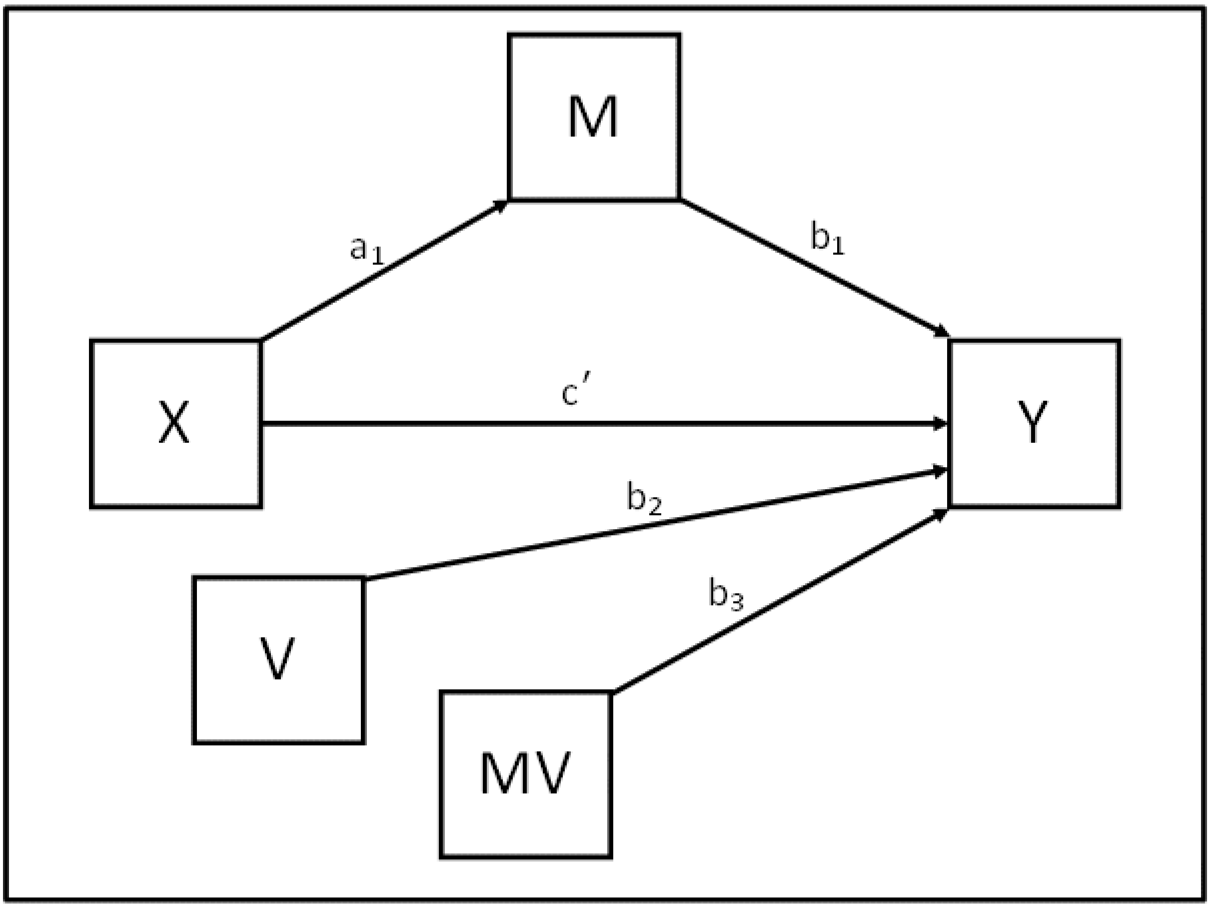
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3MV + c'X
M = a0 + a1X

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3MV + c'X
M = a0 + a1X

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X) + b2V + b3(a0 + a1X)V + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + b2V + a0b3V + a1b3XV + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + b2V + a0b3V) + (a1b1 + a1b3V + c')X

Hence...

One indirect effect(s) of X on Y, conditional on V: a1b1 + a1b3V = a1(b1 + b3V)

One direct effect of X on Y: c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - V
! Outcome variable - Y

USEVARIABLES = X M V Y MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON MV (b3);

   Y ON X (cdash);

   [M] (a0);
   M ON X (a1);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for
V
! for example, of 1 SD below mean, mean, 1 SD above mean
```

```
! 1 moderator, 3 values for it
! arbitrary naming convention for conditional indirect and
total effects used below:
! MED_Q = medium value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_V MED_V HIGH_V
    IND_LOWV IND_MEDV IND_HIV
    TOT_LOWV TOT_MEDV TOT_HIV);

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    IND_LOWV = a1*b1 + a1*b3*LOW_V;
    IND_MEDV = a1*b1 + a1*b3*MED_V;
    IND_HIV = a1*b1 + a1*b3*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TOT_LOWV = IND_LOWV + cdash;
    TOT_MEDV = IND_MEDV + cdash;
    TOT_HIV = IND_HIV + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(LOMOD MEDMOD HIMOD);

    LOOP(XVAL,1,5,0.1);

    LOMOD = IND_LOWV*XVAL;
    MEDMOD = IND_MEDV*XVAL;
    HIMOD = IND_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 15: 1 or more mediators, in parallel if multiple (example uses 1), 1 moderator of both Mediator-DV and direct IV-DV path

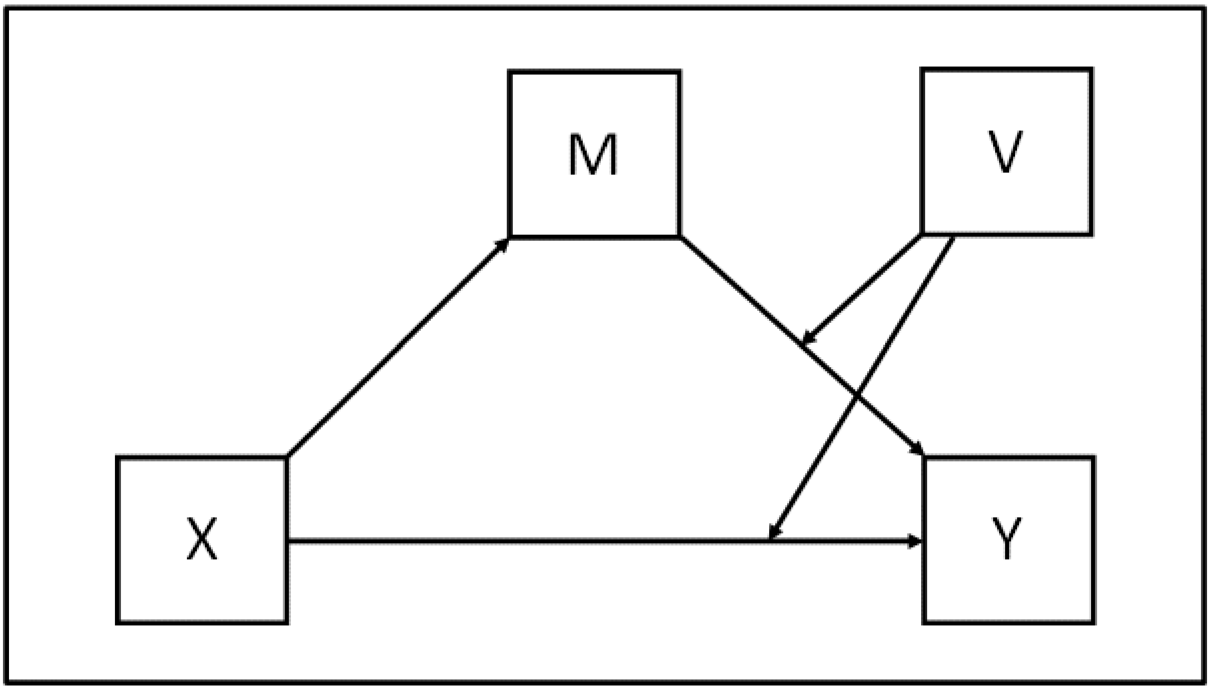
Example Variables: 1 predictor X, 1 mediator M, 1 moderator V, 1 outcome Y

Preliminary notes:

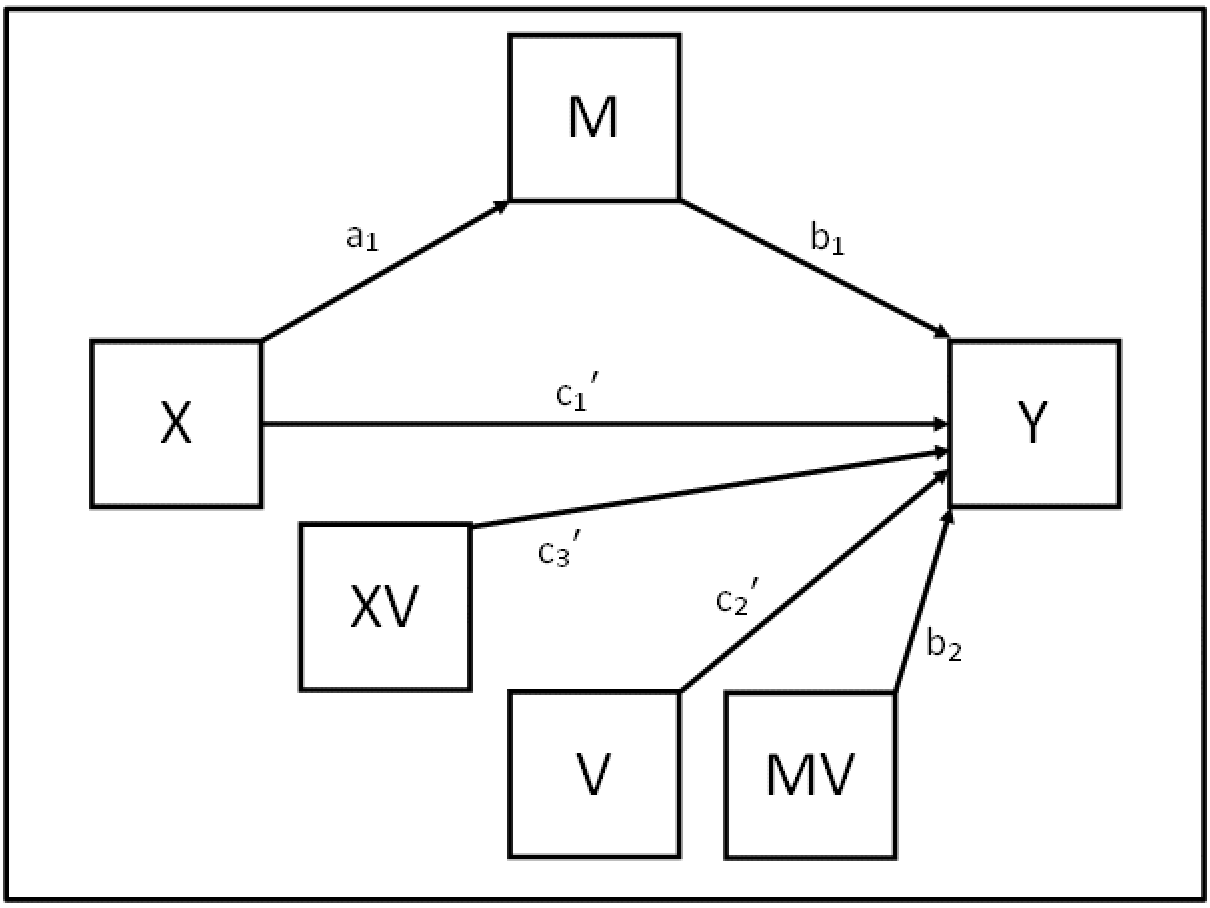
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + c1'X + c2'V + c3'XV
M = a0 + a1X

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + c1'X + c2'V + c3'XV
M = a0 + a1X

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X) + b2(a0 + a1X)V + c1'X + c2'V + c3'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a0b2V + a1b2XV + c1'X + c2'V + c3'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + c2'V + a0b2V) + (a1b1 + a1b2V + c1' + c3'V)X

Hence... One indirect effect(s) of X on Y, conditional on V:

a1b1 + a1b2V = a1(b1 + b2V)

One direct effect of X on Y, conditional on V:

c1' + c3'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - V
! Outcome variable - Y

USEVARIABLES = X M V Y MV XV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MV (b2);

   Y ON X (cdash1);
   Y ON V (cdash2);
   Y ON XV (cdash3);

   [M] (a0);
   M ON X (a1);

! Use model constraint subcommand to test conditional indirect
effects
```

```
! You need to pick low, medium and high moderator values for
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 1 moderator, 3 values for it
! arbitrary naming convention for conditional indirect and
total effects used below:
! MED_Q = medium value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_V MED_V HIGH_V
    IND_LOWV IND_MEDV IND_HIV
    DIR_LOWV DIR_MEDV DIR_HIV
    TOT_LOWV TOT_MEDV TOT_HIV);

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    IND_LOWV = a1*b1 + a1*b2*LOW_V;
    IND_MEDV = a1*b1 + a1*b2*MED_V;
    IND_HIV = a1*b1 + a1*b2*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWV = cdash1 + cdash3*LOW_V;
    DIR_MEDV = cdash1 + cdash3*MED_V;
    DIR_HIV = cdash1 + cdash3*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TOT_LOWV = IND_LOWV + DIR_LOWV;
    TOT_MEDV = IND_MEDV + DIR_MEDV;
    TOT_HIV = IND_HIV + DIR_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(LOMOD MEDMOD HIMOD);
```

```
    LOOP(XVAL,1,5,0.1);

    LOMOD = IND_LOWV*XVAL;
    MEDMOD = IND_MEDV*XVAL;
    HIMOD = IND_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 16: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the Mediator-DV path only

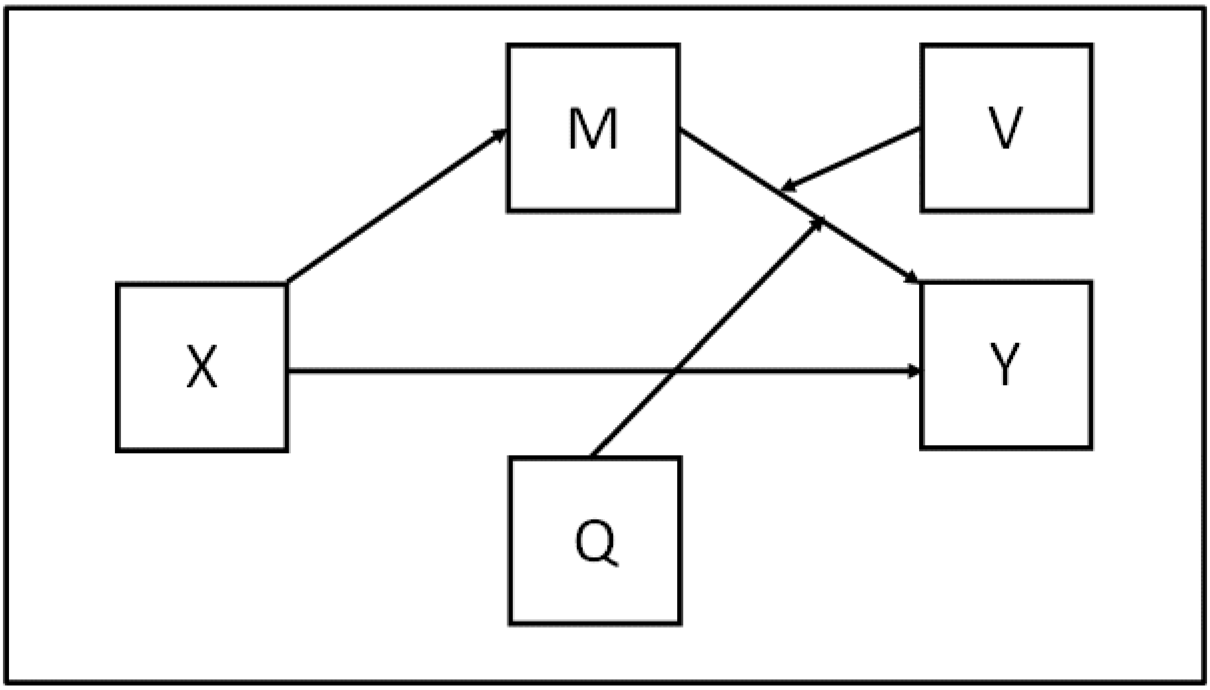
Example Variables: 1 predictor X, 1 mediator M, 2 moderators V, Q, 1 outcome Y

Preliminary notes:

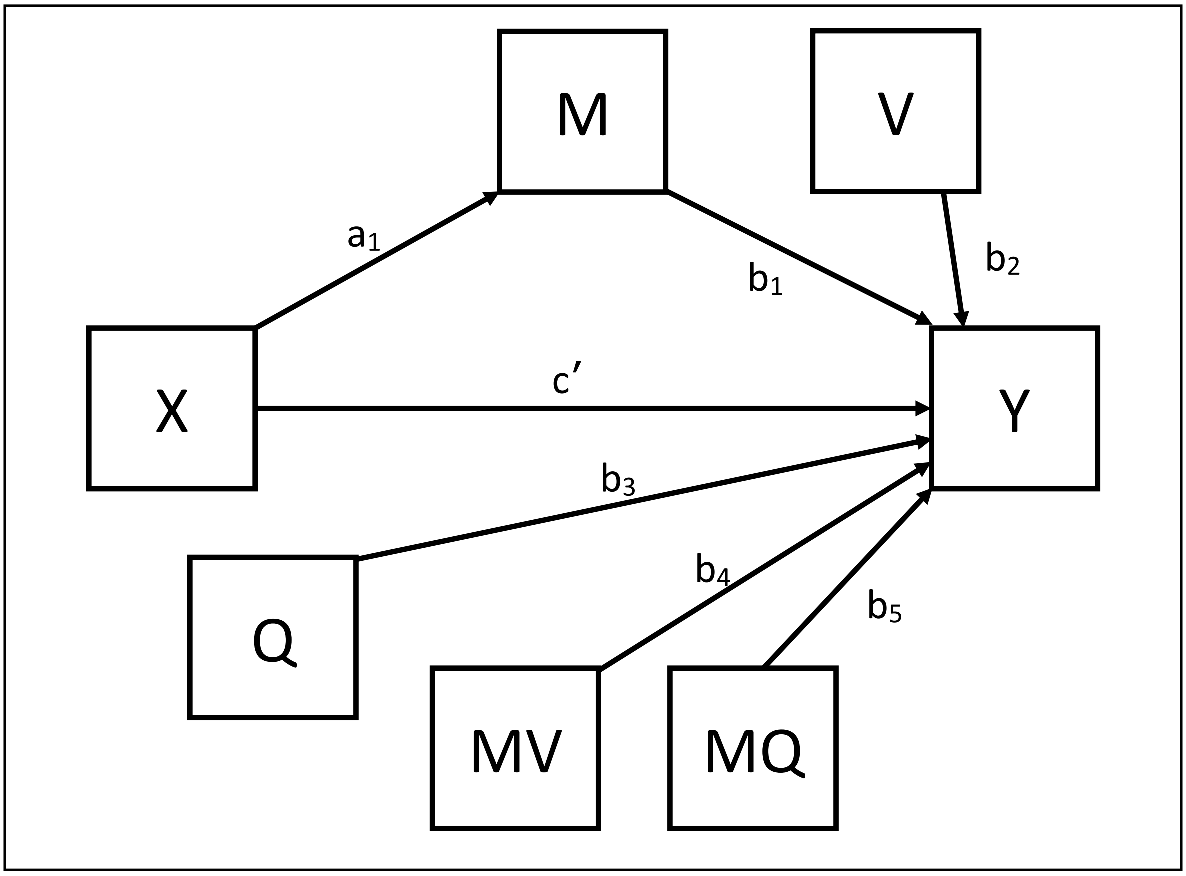
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c'X
M = a0 + a1X

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c'X
M = a0 + a1X

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X) + b2V + b3Q + b4(a0 + a1X)V + b5(a0 + a1X)Q + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + b2V + b3Q + a0b4V + a1b4XV + a0b5Q + a1b5XQ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + b2V + b3Q + a0b4V + a0b5Q) + (a1b1 + a1b4V + a1b5Q + c')X

Hence...

One indirect effect(s) of X on Y, conditional on V, Q:

a1b1 + a1b4V + a1b5Q = a1(b1 + b4V + b5Q)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - V, Q
! Outcome variable - Y

USEVARIABLES = X M V Q Y MV MQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);

   Y ON X (cdash);

   [M] (a0);
   M ON X (a1);

! Use model constraint subcommand to test conditional indirect
effects
```

```
! You need to pick low, medium and high moderator values for V,
Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_V MED_V HIGH_V LOW_Q MED_Q HIGH_Q
    ILOV_LOQ IMEV_LOQ IHIV_LOQ ILOV_MEQ IMEV_MEQ IHIV_MEQ
    ILOV_HIQ IMEV_HIQ IHIV_HIQ
    TLOV_LOQ TMEV_LOQ THIV_LOQ TLOV_MEQ TMEV_MEQ THIV_MEQ
    TLOV_HIQ TMEV_HIQ THIV_HIQ);

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILOV_LOQ = a1*b1 + a1*b4*LOW_V + a1*b5*LOW_Q;
    IMEV_LOQ = a1*b1 + a1*b4*MED_V + a1*b5*LOW_Q;
    IHIV_LOQ = a1*b1 + a1*b4*HIGH_V + a1*b5*LOW_Q;

    ILOV_MEQ = a1*b1 + a1*b4*LOW_V + a1*b5*MED_Q;
    IMEV_MEQ = a1*b1 + a1*b4*MED_V + a1*b5*MED_Q;
    IHIV_MEQ = a1*b1 + a1*b4*HIGH_V + a1*b5*MED_Q;

    ILOV_HIQ = a1*b1 + a1*b4*LOW_V + a1*b5*HIGH_Q;
    IMEV_HIQ = a1*b1 + a1*b4*MED_V + a1*b5*HIGH_Q;
    IHIV_HIQ = a1*b1 + a1*b4*HIGH_V + a1*b5*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLOV_LOQ = ILOV_LOQ + cdash;
    TMEV_LOQ = IMEV_LOQ + cdash;
    THIV_LOQ = IHIV_LOQ + cdash;
```

```
    TLOV_MEQ = ILOV_MEQ + cdash;
    TMEV_MEQ = IMEV_MEQ + cdash;
    THIV_MEQ = IHIV_MEQ + cdash;

    TLOV_HIQ = ILOV_HIQ + cdash;
    TMEV_HIQ = IMEV_HIQ + cdash;
    THIV_HIQ = IHIV_HIQ + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOV_LOQ PMEV_LOQ PHIV_LOQ PLOV_MEQ PMEV_MEQ
PHIV_MEQ
    PLOV_HIQ PMEV_HIQ PHIV_HIQ);

    LOOP(XVAL,1,5,0.1);

    PLOV_LOQ = ILOV_LOQ*XVAL;
    PMEV_LOQ = IMEV_LOQ*XVAL;
    PHIV_LOQ = IHIV_LOQ*XVAL;

    PLOV_MEQ = ILOV_MEQ*XVAL;
    PMEV_MEQ = IMEV_MEQ*XVAL;
    PHIV_MEQ = IHIV_MEQ*XVAL;

    PLOV_HIQ = ILOV_HIQ*XVAL;
    PMEV_HIQ = IMEV_HIQ*XVAL;
    PHIV_HIQ = IHIV_HIQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 17: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating both the Mediator-DV and direct IV-DV path

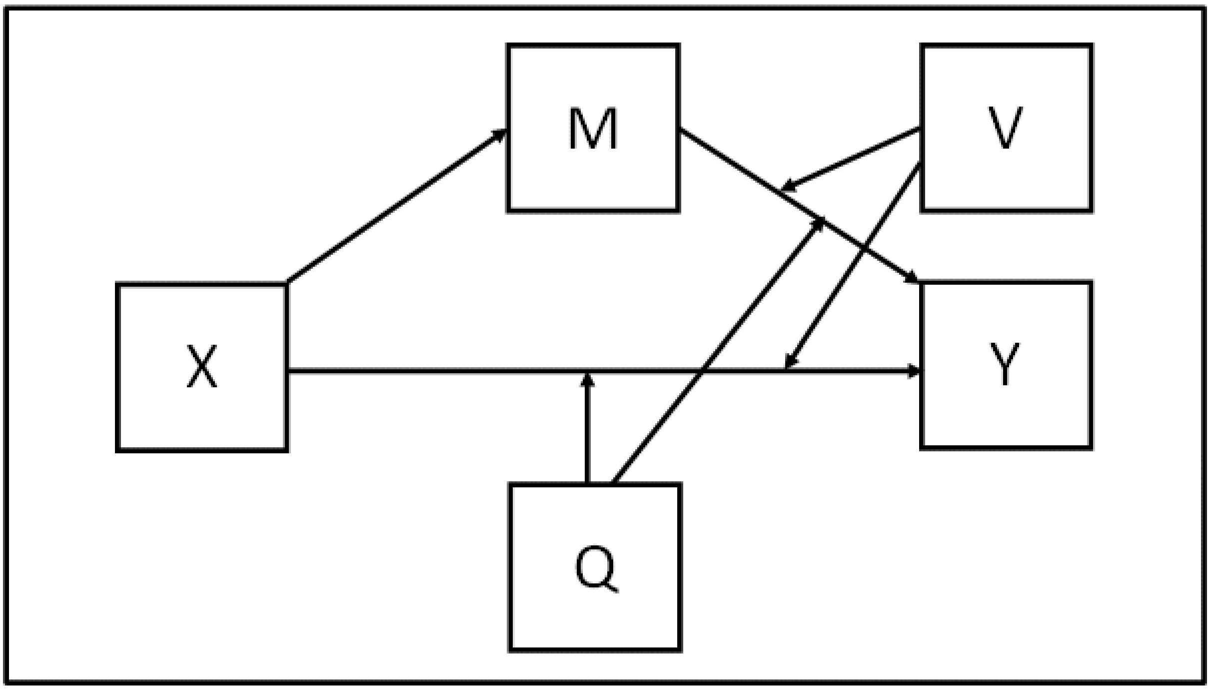
Example Variables: 1 predictor X, 1 mediator M, 2 moderators V, Q, 1 outcome Y

Preliminary notes:

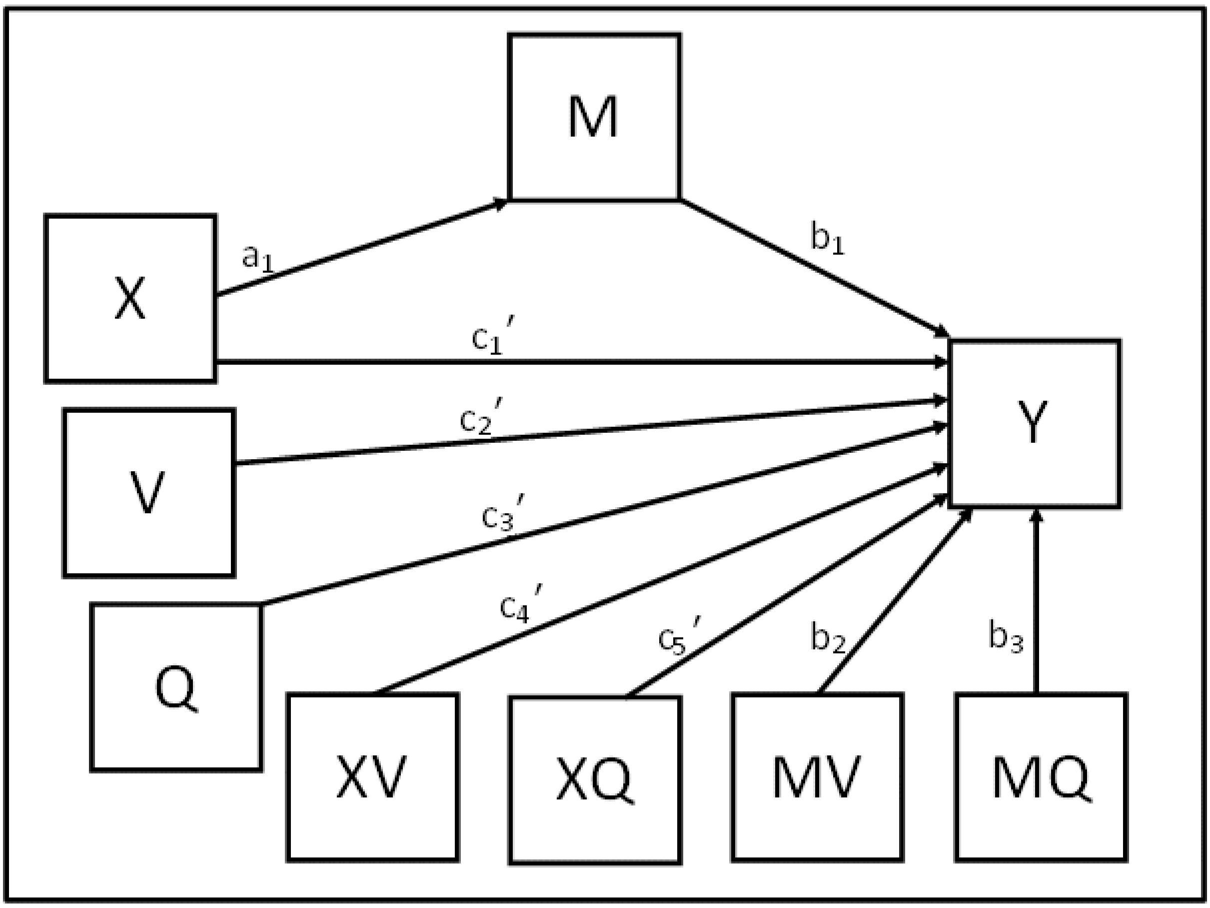
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ
M = a0 + a1X

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ
M = a0 + a1X

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X) + b2(a0 + a1X)V + b3(a0 + a1X)Q + c1'X + c2'V + c3'Q + c4'XV + c5'XQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a0b2V + a1b2XV + a0b3Q + a1b3XQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a0b2V + a0b3Q + c2'V + c3'Q) + (a1b1 + a1b2V + a1b3Q + c1' + c4'V + c5'Q)X

Hence...

One indirect effect(s) of X on Y, conditional on V, Q:

a1b1 + a1b2V + a1b3Q = a1(b1 + b2V + b3Q)

One direct effect of X on Y, conditional on V, Q:

c1' + c4'V + c5'Q

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – V, Q
! Outcome variable - Y

USEVARIABLES = X M V Q Y MV MQ XV XQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XQ = X*Q;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MV (b2);
   Y ON MQ (b3);

   Y ON X (cdash1);
   Y ON V (cdash2);
```

```
    Y ON Q (cdash3);
    Y ON XV (cdash4);
    Y ON XQ (cdash5);

    [M] (a0);
    M ON X (a1);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for V,
Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_V MED_V HIGH_V LOW_Q MED_Q HIGH_Q
    ILOV_LOQ IMEV_LOQ IHIV_LOQ ILOV_MEQ IMEV_MEQ IHIV_MEQ
    ILOV_HIQ IMEV_HIQ IHIV_HIQ
    DLOV_LOQ DMEV_LOQ DHIV_LOQ DLOV_MEQ DMEV_MEQ DHIV_MEQ
    DLOV_HIQ DMEV_HIQ DHIV_HIQ
    TLOV_LOQ TMEV_LOQ THIV_LOQ TLOV_MEQ TMEV_MEQ THIV_MEQ
    TLOV_HIQ TMEV_HIQ THIV_HIQ);

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILOV_LOQ = a1*b1 + a1*b2*LOW_V + a1*b3*LOW_Q;
    IMEV_LOQ = a1*b1 + a1*b2*MED_V + a1*b3*LOW_Q;
    IHIV_LOQ = a1*b1 + a1*b2*HIGH_V + a1*b3*LOW_Q;

    ILOV_MEQ = a1*b1 + a1*b2*LOW_V + a1*b3*MED_Q;
    IMEV_MEQ = a1*b1 + a1*b2*MED_V + a1*b3*MED_Q;
    IHIV_MEQ = a1*b1 + a1*b2*HIGH_V + a1*b3*MED_Q;
```

```
    ILOV_HIQ = a1*b1 + a1*b2*LOW_V + a1*b3*HIGH_Q;
    IMEV_HIQ = a1*b1 + a1*b2*MED_V + a1*b3*HIGH_Q;
    IHIV_HIQ = a1*b1 + a1*b2*HIGH_V + a1*b3*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOV_LOQ = cdash1 + cdash4*LOW_V + cdash5*LOW_Q;
    DMEV_LOQ = cdash1 + cdash4*MED_V + cdash5*LOW_Q;
    DHIV_LOQ = cdash1 + cdash4*HIGH_V + cdash5*LOW_Q;

    DLOV_MEQ = cdash1 + cdash4*LOW_V + cdash5*MED_Q;
    DMEV_MEQ = cdash1 + cdash4*MED_V + cdash5*MED_Q;
    DHIV_MEQ = cdash1 + cdash4*HIGH_V + cdash5*MED_Q;

    DLOV_HIQ = cdash1 + cdash4*LOW_V + cdash5*HIGH_Q;
    DMEV_HIQ = cdash1 + cdash4*MED_V + cdash5*HIGH_Q;
    DHIV_HIQ = cdash1 + cdash4*HIGH_V + cdash5*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLOV_LOQ = ILOV_LOQ + DLOV_LOQ;
    TMEV_LOQ = IMEV_LOQ + DMEV_LOQ;
    THIV_LOQ = IHIV_LOQ + DHIV_LOQ;

    TLOV_MEQ = ILOV_MEQ + DLOV_MEQ;
    TMEV_MEQ = IMEV_MEQ + DMEV_MEQ;
    THIV_MEQ = IHIV_MEQ + DHIV_MEQ;

    TLOV_HIQ = ILOV_HIQ + DLOV_HIQ;
    TMEV_HIQ = IMEV_HIQ + DMEV_HIQ;
    THIV_HIQ = IHIV_HIQ + DHIV_HIQ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOV_LOQ PMEV_LOQ PHIV_LOQ PLOV_MEQ PMEV_MEQ
PHIV_MEQ
    PLOV_HIQ PMEV_HIQ PHIV_HIQ);

    LOOP(XVAL,1,5,0.1);

    PLOV_LOQ = ILOV_LOQ*XVAL;
    PMEV_LOQ = IMEV_LOQ*XVAL;
    PHIV_LOQ = IHIV_LOQ*XVAL;
```

```
    PLOV_MEQ = ILOV_MEQ*XVAL;
    PMEV_MEQ = IMEV_MEQ*XVAL;
    PHIV_MEQ = IHIV_MEQ*XVAL;

    PLOV_HIQ = ILOV_HIQ*XVAL;
    PMEV_HIQ = IMEV_HIQ*XVAL;
    PHIV_HIQ = IHIV_HIQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 18: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the Mediator-DV path only, all 2-way and 3-way interactions

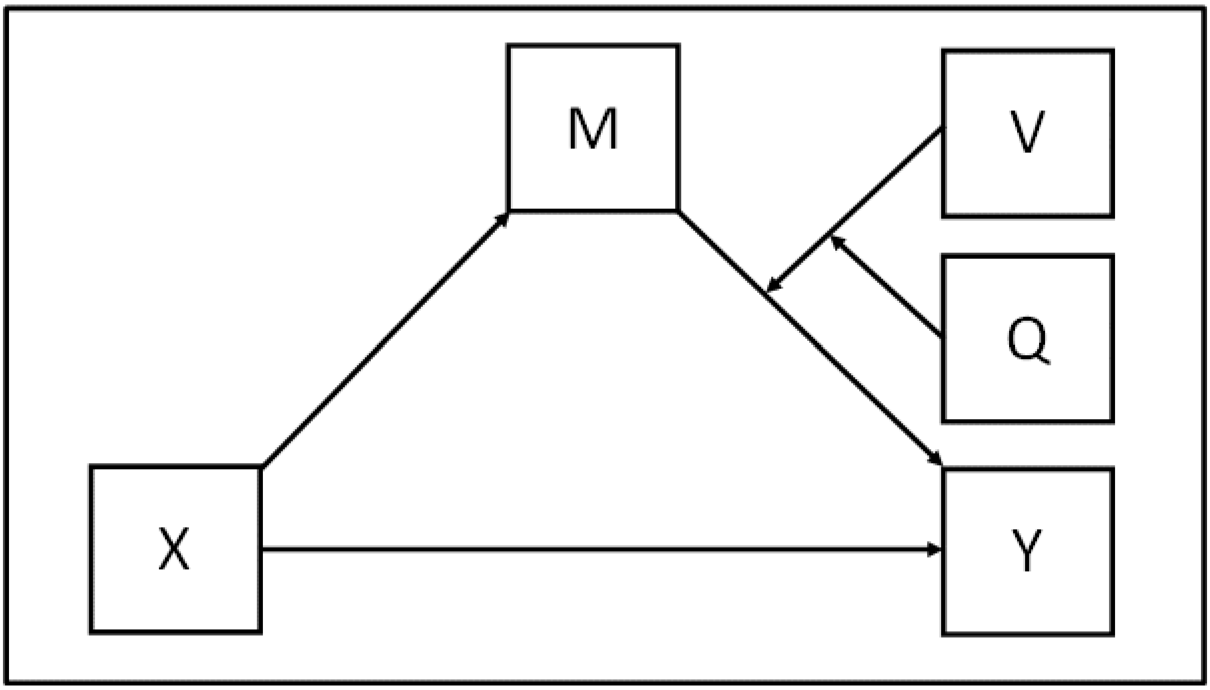
Example Variables: 1 predictor X, 1 mediator M, 2 moderators V, Q, 1 outcome Y

Preliminary notes:

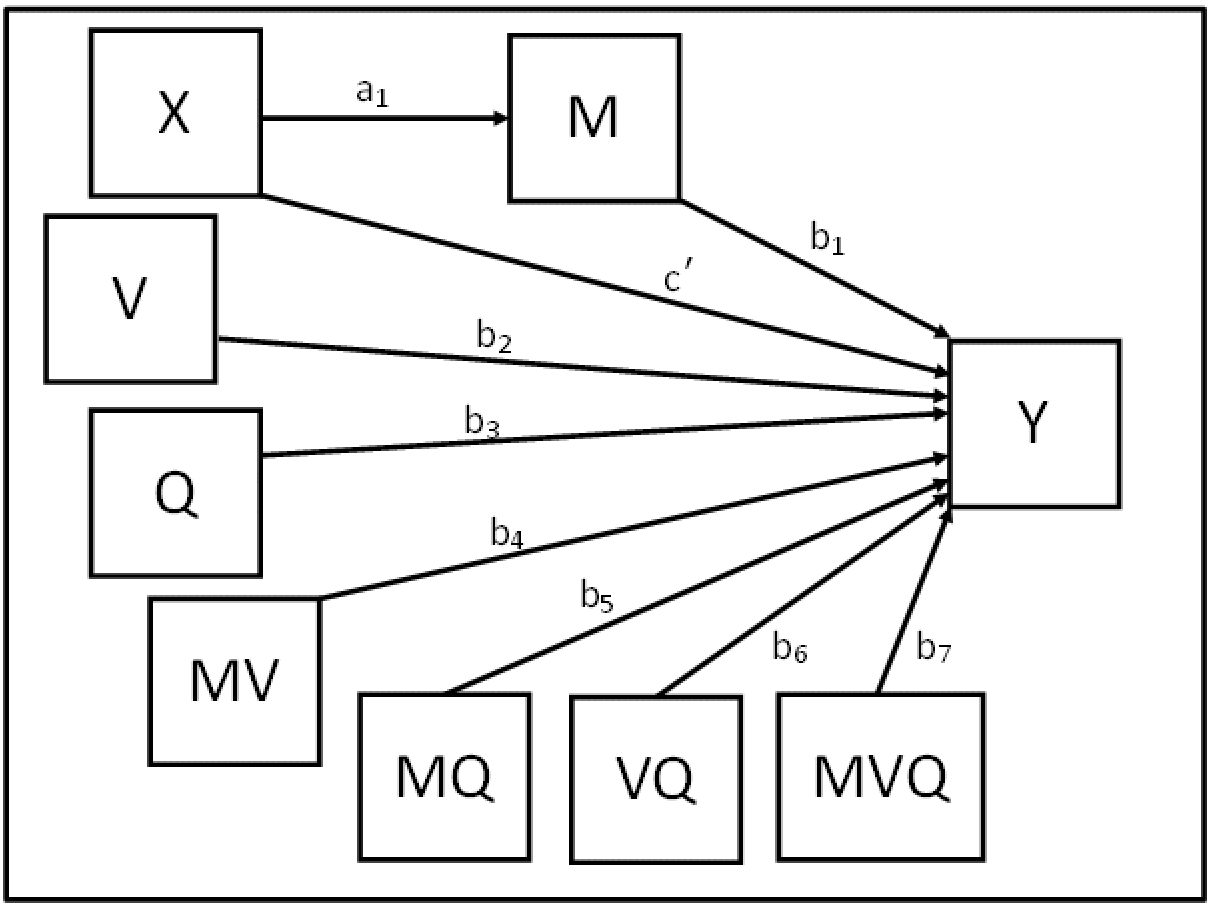
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c'X
M = a0 + a1X

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c'X
M = a0 + a1X

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X) + b2V + b3Q + b4(a0 + a1X)V + b5(a0 + a1X)Q + b6VQ + b7(a0 + a1X)VQ + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + b2V + b3Q + a0b4V + a1b4XV + a0b5Q + a1b5XQ + b6VQ + a0b7VQ + a1b7XVQ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + b2V + b3Q + a0b4V + a0b5Q + b6VQ + a0b7VQ) + (a1b1 + a1b4V + a1b5Q + a1b7VQ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on V, Q: a1b1 + a1b4V + a1b5Q + a1b7VQ = a1(b1 + b4V + b5Q + b7VQ)

One direct effect of X on Y: c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - V, Q
! Outcome variable - Y

USEVARIABLES = X M V Q Y MV MQ VQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   VQ = V*Q;
   MVQ = M*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);
   Y ON VQ (b6);
   Y ON MVQ (b7);

   Y ON X (cdash);
```

```
   [M] (a0);
   M ON X (a1);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for V,
Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_V MED_V HIGH_V LOW_Q MED_Q HIGH_Q
    ILOV_LOQ IMEV_LOQ IHIV_LOQ ILOV_MEQ IMEV_MEQ IHIV_MEQ
    ILOV_HIQ IMEV_HIQ IHIV_HIQ
    TLOV_LOQ TMEV_LOQ THIV_LOQ TLOV_MEQ TMEV_MEQ THIV_MEQ
    TLOV_HIQ TMEV_HIQ THIV_HIQ);

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILOV_LOQ = a1*b1 + a1*b4*LOW_V + a1*b5*LOW_Q +
a1*b7*LOW_V*LOW_Q;
    IMEV_LOQ = a1*b1 + a1*b4*MED_V + a1*b5*LOW_Q +
a1*b7*MED_V*LOW_Q;
    IHIV_LOQ = a1*b1 + a1*b4*HIGH_V + a1*b5*LOW_Q +
a1*b7*HIGH_V*LOW_Q;

    ILOV_MEQ = a1*b1 + a1*b4*LOW_V + a1*b5*MED_Q +
a1*b7*LOW_V*MED_Q;
    IMEV_MEQ = a1*b1 + a1*b4*MED_V + a1*b5*MED_Q +
a1*b7*MED_V*MED_Q;
    IHIV_MEQ = a1*b1 + a1*b4*HIGH_V + a1*b5*MED_Q +
a1*b7*HIGH_V*MED_Q;
```

```
    ILOV_HIQ = a1*b1 + a1*b4*LOW_V + a1*b5*HIGH_Q +
a1*b7*LOW_V*HIGH_Q;
    IMEV_HIQ = a1*b1 + a1*b4*MED_V + a1*b5*HIGH_Q +
a1*b7*MED_V*HIGH_Q;
    IHIV_HIQ = a1*b1 + a1*b4*HIGH_V + a1*b5*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLOV_LOQ = ILOV_LOQ + cdash;
    TMEV_LOQ = IMEV_LOQ + cdash;
    THIV_LOQ = IHIV_LOQ + cdash;

    TLOV_MEQ = ILOV_MEQ + cdash;
    TMEV_MEQ = IMEV_MEQ + cdash;
    THIV_MEQ = IHIV_MEQ + cdash;

    TLOV_HIQ = ILOV_HIQ + cdash;
    TMEV_HIQ = IMEV_HIQ + cdash;
    THIV_HIQ = IHIV_HIQ + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOV_LOQ PMEV_LOQ PHIV_LOQ PLOV_MEQ PMEV_MEQ
PHIV_MEQ
    PLOV_HIQ PMEV_HIQ PHIV_HIQ);

    LOOP(XVAL,1,5,0.1);

    PLOV_LOQ = ILOV_LOQ*XVAL;
    PMEV_LOQ = IMEV_LOQ*XVAL;
    PHIV_LOQ = IHIV_LOQ*XVAL;

    PLOV_MEQ = ILOV_MEQ*XVAL;
    PMEV_MEQ = IMEV_MEQ*XVAL;
    PHIV_MEQ = IHIV_MEQ*XVAL;

    PLOV_HIQ = ILOV_HIQ*XVAL;
    PMEV_HIQ = IMEV_HIQ*XVAL;
    PHIV_HIQ = IHIV_HIQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 19: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators both moderating the Mediator-DV path and direct IV-DV path, all 2-way and 3-way interactions

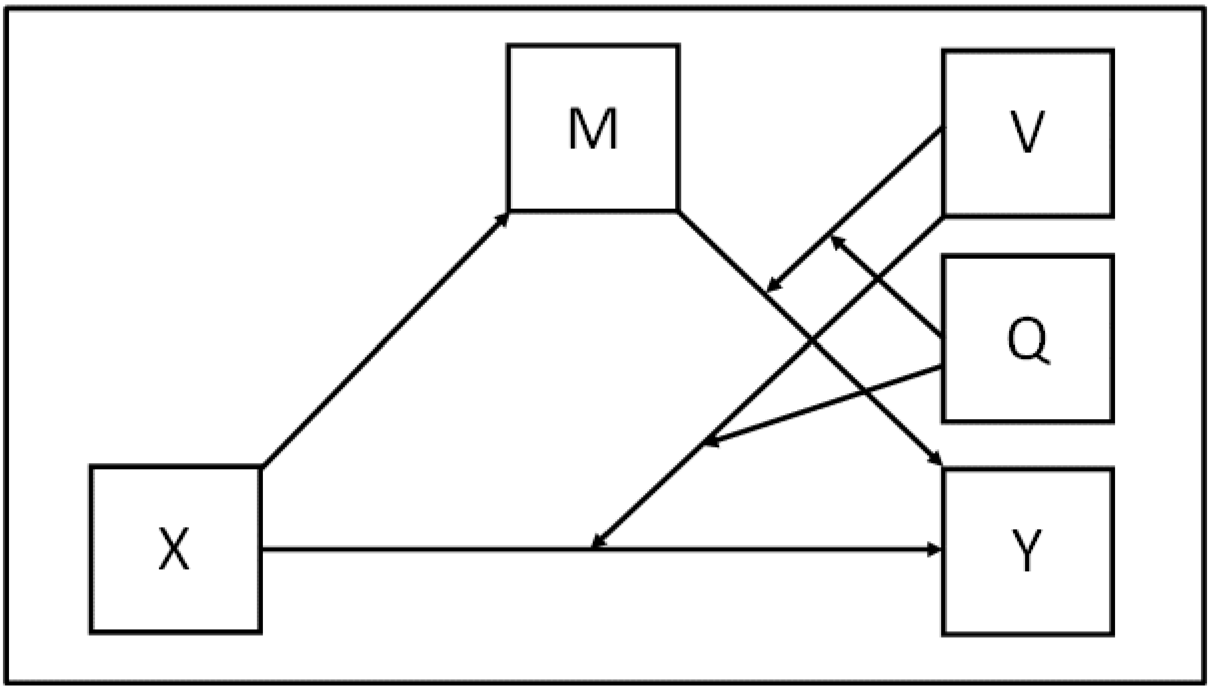
Example Variables: 1 predictor X, 1 mediator M, 2 moderators V, Q, 1 outcome Y

Preliminary notes:

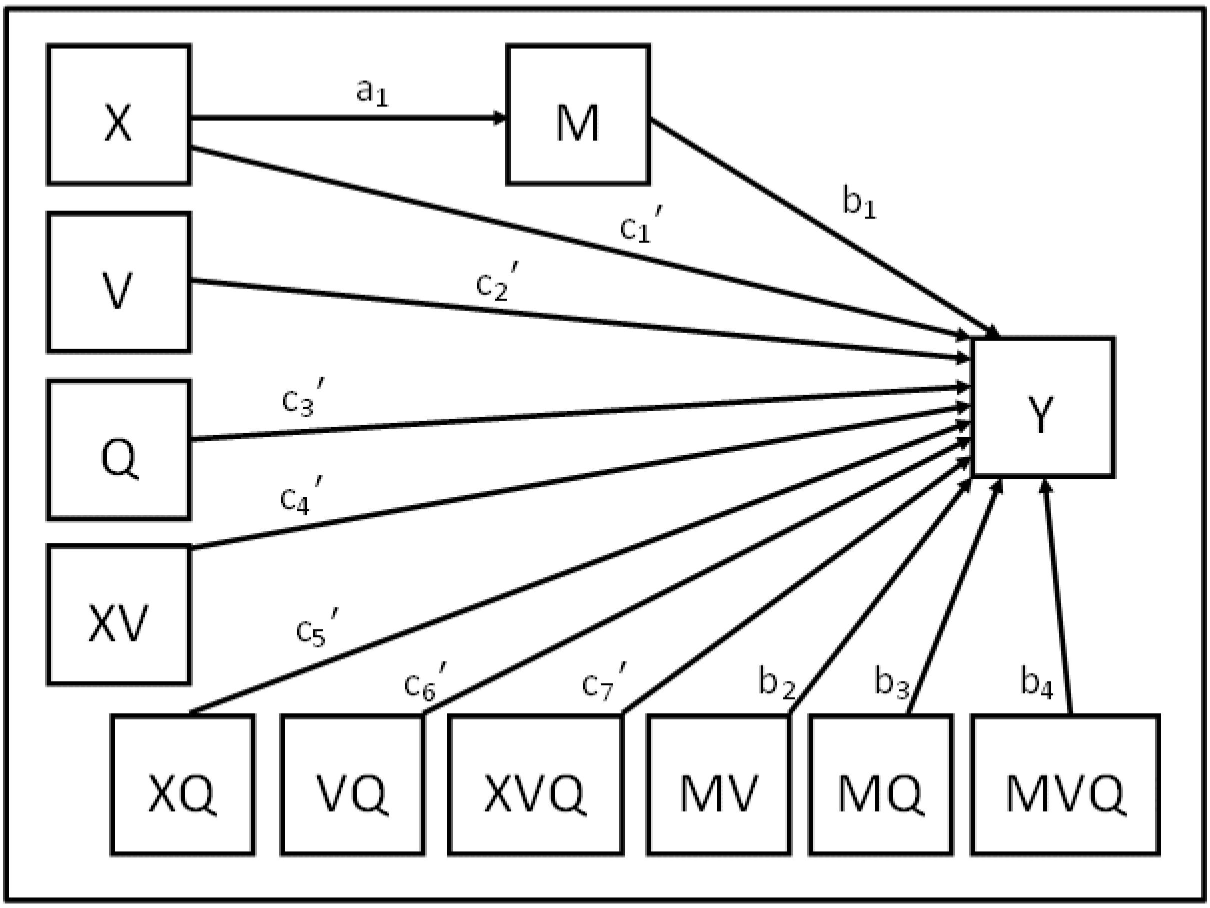
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ
M = a0 + a1X

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ
M = a0 + a1X

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X) + b2(a0 + a1X)V + b3(a0 + a1X)Q + b4(a0 + a1X)VQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a0b2V + a1b2XV + a0b3Q + a1b3XQ + a0b4VQ + a1b4XVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a0b2V + a0b3Q + a0b4VQ + c2'V + c3'Q + c6'VQ) + (a1b1 + a1b2V + a1b3Q + a1b4VQ + c1' + c4'V + c5'Q + c7'VQ)X

Hence...

One indirect effect(s) of X on Y, conditional on V, Q:

a1b1 + a1b2V + a1b3Q + a1b4VQ = a1(b1 + b2V + b3Q + b4VQ)

One direct effect of X on Y, conditional on V, Q:

c1' + c4'V + c5'Q + c7'VQ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – V, Q
! Outcome variable - Y

USEVARIABLES = X M V Q Y MV MQ XV XQ VQ MVQ XVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XQ = X*Q;
   XV = X*V;
   VQ = V*Q;
   MVQ = M*V*Q;
   XVQ = X*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
```

```
    Y ON M (b1);
    Y ON MV (b2);
    Y ON MQ (b3);
    Y ON MVQ (b4);

    Y ON X (cdash1);
    Y ON V (cdash2);
    Y ON Q (cdash3);
    Y ON XV (cdash4);
    Y ON XQ (cdash5);
    Y ON VQ (cdash6);
    Y ON XVQ (cdash7);

    [M] (a0);
    M ON X (a1);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for V,
Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_V MED_V HIGH_V LOW_Q MED_Q HIGH_Q
    ILOV_LOQ IMEV_LOQ IHIV_LOQ ILOV_MEQ IMEV_MEQ IHIV_MEQ
    ILOV_HIQ IMEV_HIQ IHIV_HIQ
    DLOV_LOQ DMEV_LOQ DHIV_LOQ DLOV_MEQ DMEV_MEQ DHIV_MEQ
    DLOV_HIQ DMEV_HIQ DHIV_HIQ
    TLOV_LOQ TMEV_LOQ THIV_LOQ TLOV_MEQ TMEV_MEQ THIV_MEQ
    TLOV_HIQ TMEV_HIQ THIV_HIQ);

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q
```

```
! Calc conditional indirect effects for each combination of
moderator values

    ILOV_LOQ = a1*b1 + a1*b2*LOW_V + a1*b3*LOW_Q +
a1*b4*LOW_V*LOW_Q;
    IMEV_LOQ = a1*b1 + a1*b2*MED_V + a1*b3*LOW_Q +
a1*b4*MED_V*LOW_Q;
    IHIV_LOQ = a1*b1 + a1*b2*HIGH_V + a1*b3*LOW_Q +
a1*b4*HIGH_V*LOW_Q;

    ILOV_MEQ = a1*b1 + a1*b2*LOW_V + a1*b3*MED_Q +
a1*b4*LOW_V*MED_Q;
    IMEV_MEQ = a1*b1 + a1*b2*MED_V + a1*b3*MED_Q +
a1*b4*MED_V*MED_Q;
    IHIV_MEQ = a1*b1 + a1*b2*HIGH_V + a1*b3*MED_Q +
a1*b4*HIGH_V*MED_Q;

    ILOV_HIQ = a1*b1 + a1*b2*LOW_V + a1*b3*HIGH_Q +
a1*b4*LOW_V*HIGH_Q;
    IMEV_HIQ = a1*b1 + a1*b2*MED_V + a1*b3*HIGH_Q +
a1*b4*MED_V*HIGH_Q;
    IHIV_HIQ = a1*b1 + a1*b2*HIGH_V + a1*b3*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOV_LOQ = cdash1 + cdash4*LOW_V + cdash5*LOW_Q +
cdash7*LOW_V*LOW_Q;
    DMEV_LOQ = cdash1 + cdash4*MED_V + cdash5*LOW_Q +
cdash7*MED_V*LOW_Q;
    DHIV_LOQ = cdash1 + cdash4*HIGH_V + cdash5*LOW_Q +
cdash7*HIGH_V*LOW_Q;

    DLOV_MEQ = cdash1 + cdash4*LOW_V + cdash5*MED_Q +
cdash7*LOW_V*MED_Q;
    DMEV_MEQ = cdash1 + cdash4*MED_V + cdash5*MED_Q +
cdash7*MED_V*MED_Q;
    DHIV_MEQ = cdash1 + cdash4*HIGH_V + cdash5*MED_Q +
cdash7*HIGH_V*MED_Q;

    DLOV_HIQ = cdash1 + cdash4*LOW_V + cdash5*HIGH_Q +
cdash7*LOW_V*HIGH_Q;
    DMEV_HIQ = cdash1 + cdash4*MED_V + cdash5*HIGH_Q +
cdash7*MED_V*HIGH_Q;
    DHIV_HIQ = cdash1 + cdash4*HIGH_V + cdash5*HIGH_Q +
cdash7*HIGH_V*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values
```

```
    TLOV_LOQ = ILOV_LOQ + DLOV_LOQ;
    TMEV_LOQ = IMEV_LOQ + DMEV_LOQ;
    THIV_LOQ = IHIV_LOQ + DHIV_LOQ;

    TLOV_MEQ = ILOV_MEQ + DLOV_MEQ;
    TMEV_MEQ = IMEV_MEQ + DMEV_MEQ;
    THIV_MEQ = IHIV_MEQ + DHIV_MEQ;

    TLOV_HIQ = ILOV_HIQ + DLOV_HIQ;
    TMEV_HIQ = IMEV_HIQ + DMEV_HIQ;
    THIV_HIQ = IHIV_HIQ + DHIV_HIQ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOV_LOQ PMEV_LOQ PHIV_LOQ PLOV_MEQ PMEV_MEQ
PHIV_MEQ
    PLOV_HIQ PMEV_HIQ PHIV_HIQ);

    LOOP(XVAL,1,5,0.1);

    PLOV_LOQ = ILOV_LOQ*XVAL;
    PMEV_LOQ = IMEV_LOQ*XVAL;
    PHIV_LOQ = IHIV_LOQ*XVAL;

    PLOV_MEQ = ILOV_MEQ*XVAL;
    PMEV_MEQ = IMEV_MEQ*XVAL;
    PHIV_MEQ = IHIV_MEQ*XVAL;

    PLOV_HIQ = ILOV_HIQ*XVAL;
    PMEV_HIQ = IMEV_HIQ*XVAL;
    PHIV_HIQ = IHIV_HIQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 20: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating the Mediator- DV path, 3-way interaction, 1 also moderating direct IV-DV path

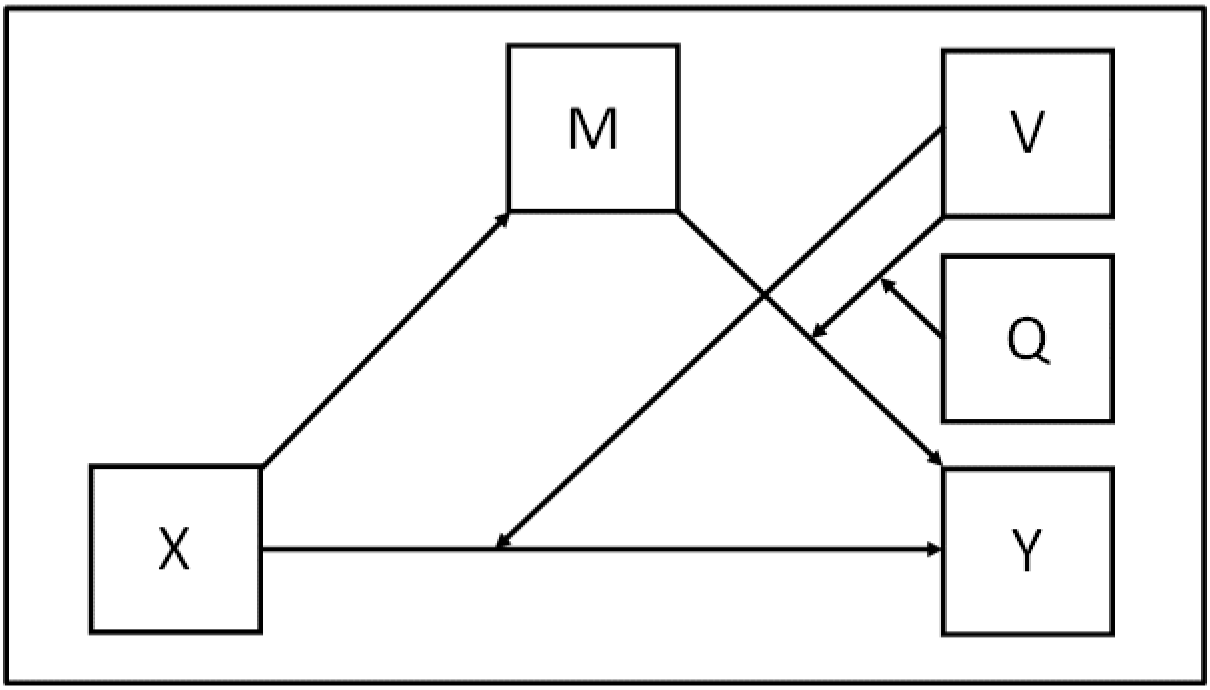
Example Variables: 1 predictor X, 1 mediator M, 2 moderators V, Q, 1 outcome Y

Preliminary notes:

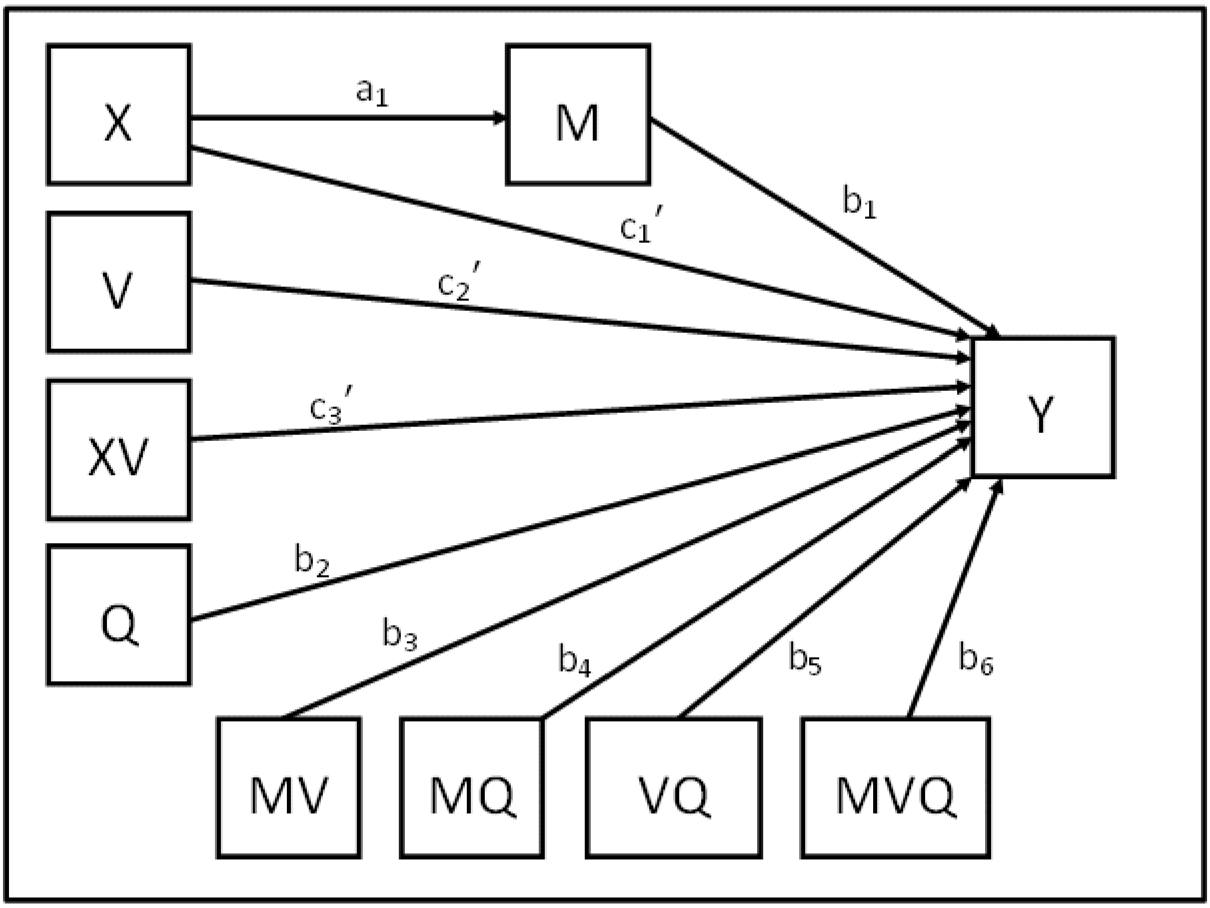
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2Q + b3MV + b4MQ + b5VQ + b6MVQ + c1'X + c2'V + c3'XV
M = a0 + a1X

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2Q + b3MV + b4MQ + b5VQ + b6MVQ + c1'X + c2'V + c3'XV
M = a0 + a1X

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X) + b2Q + b3(a0 + a1X)V + b4(a0 + a1X)Q + b5VQ + b6(a0 + a1X)VQ + c1'X + c2'V + c3'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + b2Q + a0b3V + a1b3XV + a0b4Q + a1b4XQ + b5VQ + a0b6VQ + a1b6XVQ + c1'X + c2'V + c'3XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + b2Q + a0b3V + a0b4Q + b5VQ + a0b6VQ + c2'V) + (a1b1 + a1b3V + a1b4Q + a1b6VQ + c1' + c3'V)X

Hence...

One indirect effect(s) of X on Y, conditional on V, Q:

a1b1 + a1b3V + a1b4Q + a1b6VQ = a1(b1 + b3V + b4Q + b6VQ)

One direct effect of X on Y, conditional on V:

c1' + c3'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - V, Q
! Outcome variable - Y

USEVARIABLES = X M V Q Y MV MQ XV VQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XV = X*V;
   VQ = V*Q;
   MVQ = M*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON Q (b2);
   Y ON MV (b3);
   Y ON MQ (b4);
```

```
    Y ON VQ (b5);
    Y ON MVQ (b6);

    Y ON X (cdash1);
    Y ON V (cdash2);
    Y ON XV (cdash3);

    [M] (a0);
    M ON X (a1);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for V,
Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_V MED_V HIGH_V LOW_Q MED_Q HIGH_Q
    ILOV_LOQ IMEV_LOQ IHIV_LOQ ILOV_MEQ IMEV_MEQ IHIV_MEQ
    ILOV_HIQ IMEV_HIQ IHIV_HIQ
    DIR_LOWV DIR_MEDV DIR_HIV
    TLOV_LOQ TMEV_LOQ THIV_LOQ TLOV_MEQ TMEV_MEQ THIV_MEQ
    TLOV_HIQ TMEV_HIQ THIV_HIQ);

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILOV_LOQ = a1*b1 + a1*b3*LOW_V + a1*b4*LOW_Q +
a1*b6*LOW_V*LOW_Q;
    IMEV_LOQ = a1*b1 + a1*b3*MED_V + a1*b4*LOW_Q +
a1*b6*MED_W*LOW_Q;
```

```
    IHIV_LOQ = a1*b1 + a1*b3*HIGH_V + a1*b4*LOW_Q +
a1*b6*HIGH_V*LOW_Q;

    ILOV_MEQ = a1*b1 + a1*b3*LOW_V + a1*b4*MED_Q +
a1*b6*LOW_V*MED_Q;
    IMEV_MEQ = a1*b1 + a1*b3*MED_V + a1*b4*MED_Q +
a1*b6*MED_V*MED_Q;
    IHIV_MEQ = a1*b1 + a1*b3*HIGH_V + a1*b4*MED_Q +
a1*b6*HIGH_V*MED_Q;

    ILOV_HIQ = a1*b1 + a1*b3*LOW_V + a1*b4*HIGH_Q +
a1*b6*LOW_V*HIGH_Q;
    IMEV_HIQ = a1*b1 + a1*b3*MED_V + a1*b4*HIGH_Q +
a1*b6*MED_V*HIGH_Q;
    IHIV_HIQ = a1*b1 + a1*b3*HIGH_V + a1*b4*HIGH_Q +
a1*b6*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWV = cdash1 + cdash3*LOW_V;
    DIR_MEDV = cdash1 + cdash3*MED_V;
    DIR_HIV = cdash1 + cdash3*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLOV_LOQ = ILOV_LOQ + DIR_LOWV;
    TMEV_LOQ = IMEV_LOQ + DIR_MEDV;
    THIV_LOQ = IHIV_LOQ + DIR_HIV;

    TLOV_MEQ = ILOV_MEQ + DIR_LOWV;
    TMEV_MEQ = IMEV_MEQ + DIR_MEDV;
    THIV_MEQ = IHIV_MEQ + DIR_HIV;

    TLOV_HIQ = ILOV_HIQ + DIR_LOWV;
    TMEV_HIQ = IMEV_HIQ + DIR_MEDV;
    THIV_HIQ = IHIV_HIQ + DIR_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOV_LOQ PMEV_LOQ PHIV_LOQ PLOV_MEQ PMEV_MEQ
PHIV_MEQ
    PLOV_HIQ PMEV_HIQ PHIV_HIQ);

    LOOP(XVAL,1,5,0.1);
```

```
    PLOV_LOQ = ILOV_LOQ*XVAL;
    PMEV_LOQ = IMEV_LOQ*XVAL;
    PHIV_LOQ = IHIV_LOQ*XVAL;

    PLOV_MEQ = ILOV_MEQ*XVAL;
    PMEV_MEQ = IMEV_MEQ*XVAL;
    PHIV_MEQ = IHIV_MEQ*XVAL;

    PLOV_HIQ = ILOV_HIQ*XVAL;
    PMEV_HIQ = IMEV_HIQ*XVAL;
    PHIV_HIQ = IHIV_HIQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 21: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, 1 moderating the IV-Mediator path, 1 moderating the Mediator-DV path

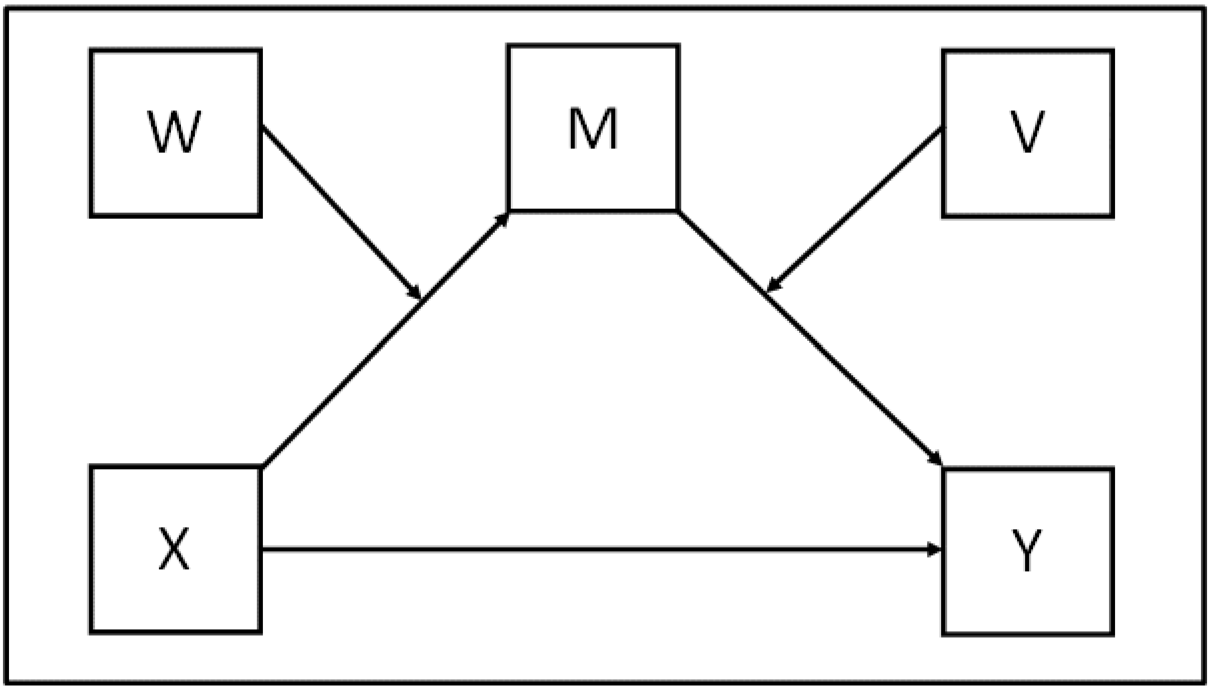
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

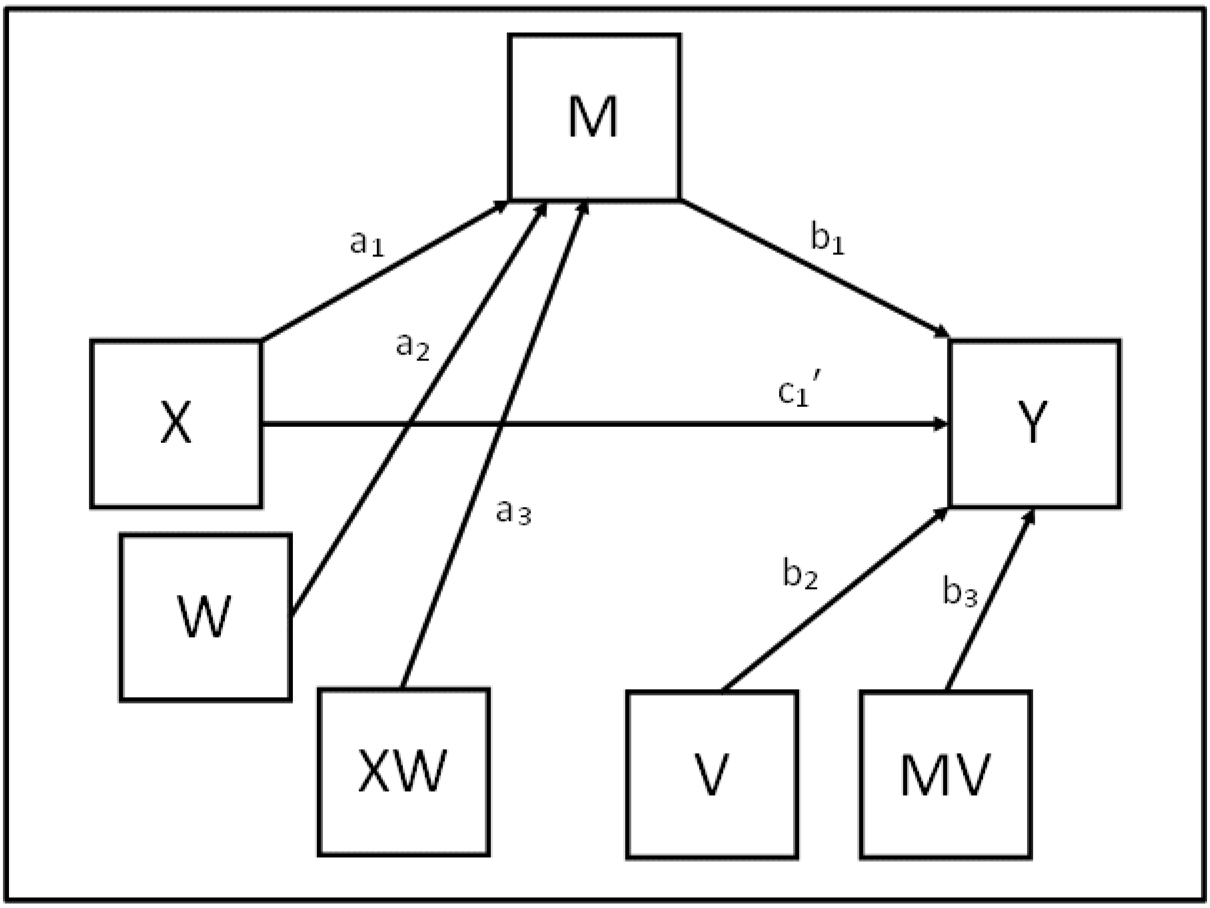
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3MV + c'X
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3MV + c'X
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2V + b3(a0 + a1X + a2W + a3XW)V + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2V + a0b3V + a1b3XV + a2b3VW + a3b3XWV + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2V + a0b3V + a2b3VW) + (a1b1 + a3b1W + a1b3V + a3b3WV + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b3V + a3b3WV = (a1 + a3W)(b1 + b3V)

One direct effect of X on Y: c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON MV (b3);

   Y ON X (cdash);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);
```

```
! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
    ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
    ILOW_HIV IMEW_HIV IHIW_HIV
    TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
    TLOW_HIV TMEW_HIV THIW_HIV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_V +
a3*b3*LOW_W*LOW_V;
    IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b3*LOW_V +
a3*b3*MED_W*LOW_V;
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b3*LOW_V +
a3*b3*HIGH_W*LOW_V;

    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b3*MED_V +
a3*b3*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b3*MED_V +
a3*b3*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b3*MED_V +
a3*b3*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b3*HIGH_V +
a3*b3*LOW_W*HIGH_V;
```

```
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b3*HIGH_V +
a3*b3*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_V +
a3*b3*HIGH_W*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + cdash;
    TMEW_LOV = IMEW_LOV + cdash;
    THIW_LOV = IHIW_LOV + cdash;

    TLOW_MEV = ILOW_MEV + cdash;
    TMEW_MEV = IMEW_MEV + cdash;
    THIW_MEV = IHIW_MEV + cdash;

    TLOW_HIV = ILOW_HIV + cdash;
    TMEW_HIV = IMEW_HIV + cdash;
    THIW_HIV = IHIW_HIV + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;

    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 22: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, 1 moderating the IV-Mediator path and direct IV-DV path, 1 moderating the Mediator-DV path

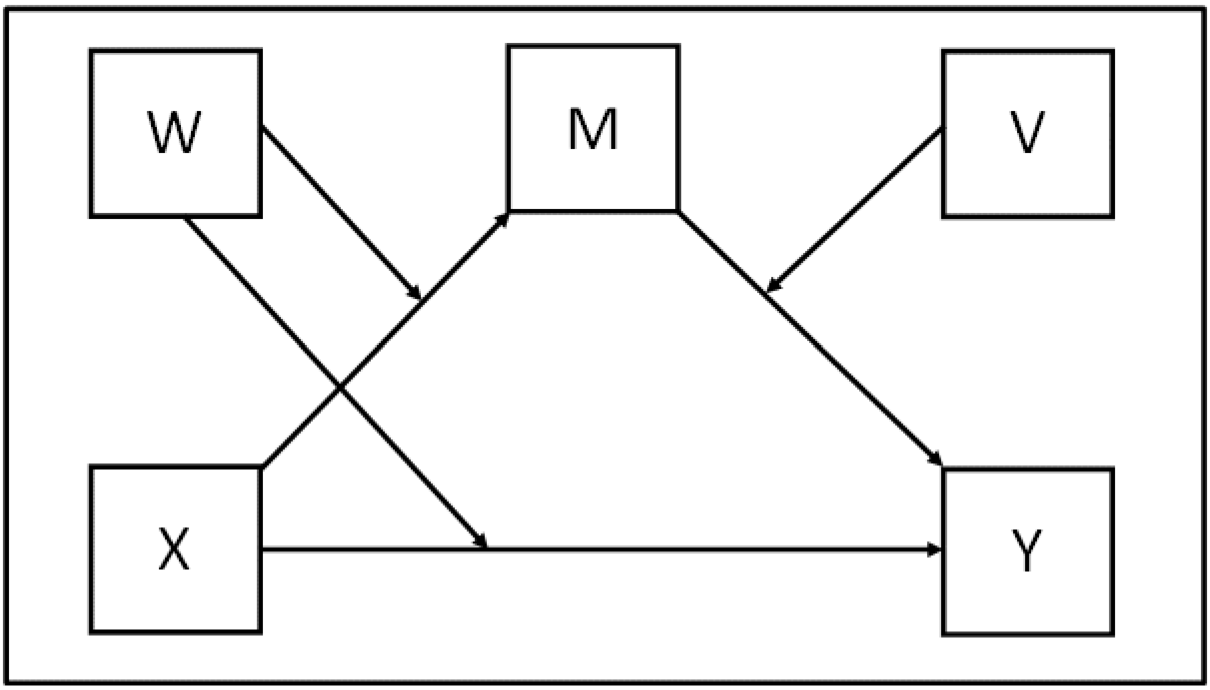
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

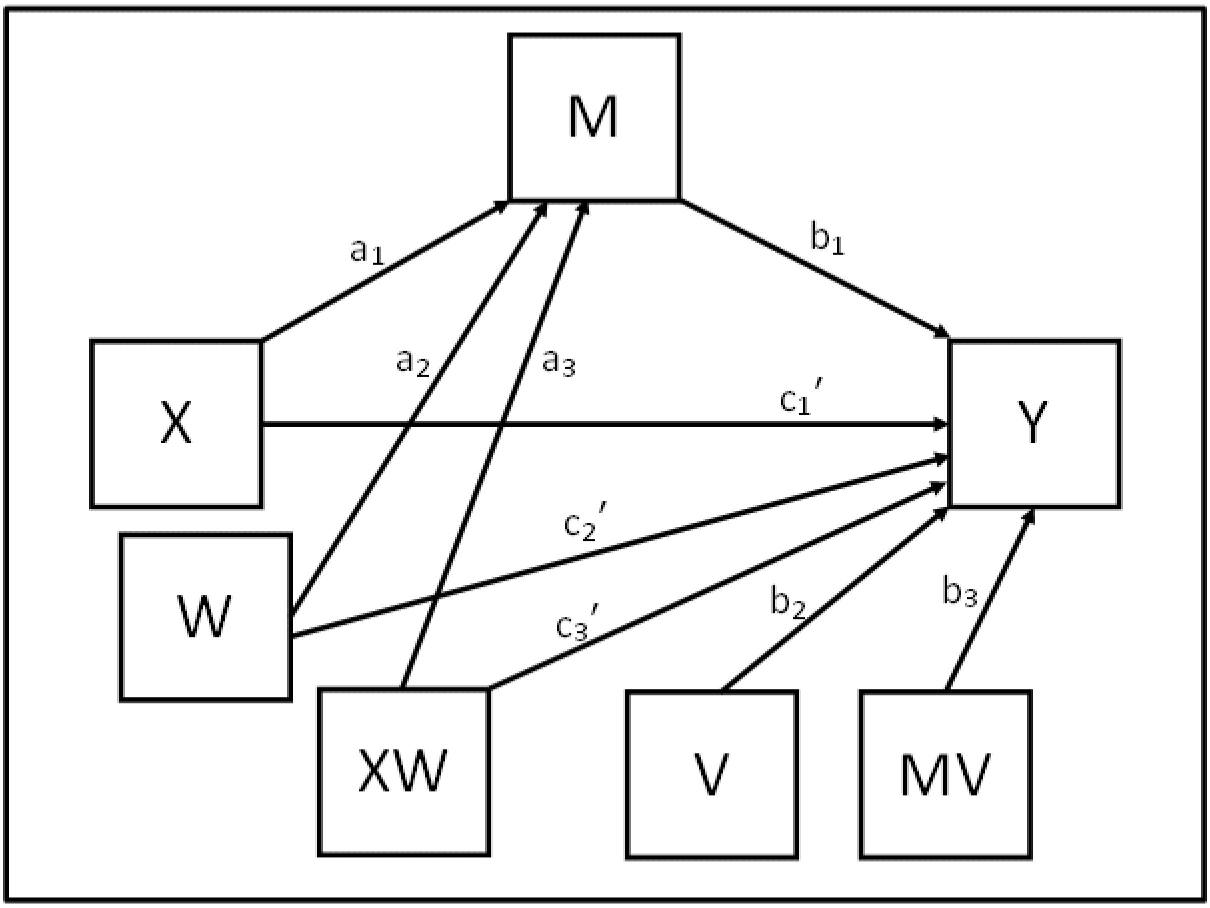
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3MV + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3MV + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2V + b3(a0 + a1X + a2W + a3XW)V + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2V + a0b3V + a1b3XV + a2b3VW + a3b3XWV + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2V + a0b3V + a2b3VW + c2'W) + (a1b1 + a3b1W + a1b3V + a3b3WV + c1' + c3'W)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b3V + a3b3WV = (a1 + a3W)(b1 + b3V)

One direct effect of X on Y, conditional on W:

c1' + c3'W

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON MV (b3);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);
```

```
    [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
    ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
    ILOW_HIV IMEW_HIV IHIW_HIV
    DIR_LOWW DIR_MEDW DIR_HIW
    TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
    TLOW_HIV TMEW_HIV THIW_HIV);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_V +
a3*b3*LOW_W*LOW_V;
    IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b3*LOW_V +
a3*b3*MED_W*LOW_V;
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b3*LOW_V +
a3*b3*HIGH_W*LOW_V;

    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b3*MED_V +
a3*b3*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b3*MED_V +
```

```
a3*b3*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b3*MED_V +
a3*b3*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b3*HIGH_V +
a3*b3*LOW_W*HIGH_V;
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b3*HIGH_V +
a3*b3*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_V +
a3*b3*HIGH_W*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWW = cdash1 + cdash3*LOW_W;
    DIR_MEDW = cdash1 + cdash3*MED_W;
    DIR_HIW = cdash1 + cdash3*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + DIR_LOWW;
    TMEW_LOV = IMEW_LOV + DIR_MEDW;
    THIW_LOV = IHIW_LOV + DIR_HIW;

    TLOW_MEV = ILOW_MEV + DIR_LOWW;
    TMEW_MEV = IMEW_MEV + DIR_MEDW;
    THIW_MEV = IHIW_MEV + DIR_HIW;

    TLOW_HIV = ILOW_HIV + DIR_LOWW;
    TMEW_HIV = IMEW_HIV + DIR_MEDW;
    THIW_HIV = IHIW_HIV + DIR_HIW;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;
```

```
    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 23: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path, 1 moderating the Mediator-DV path

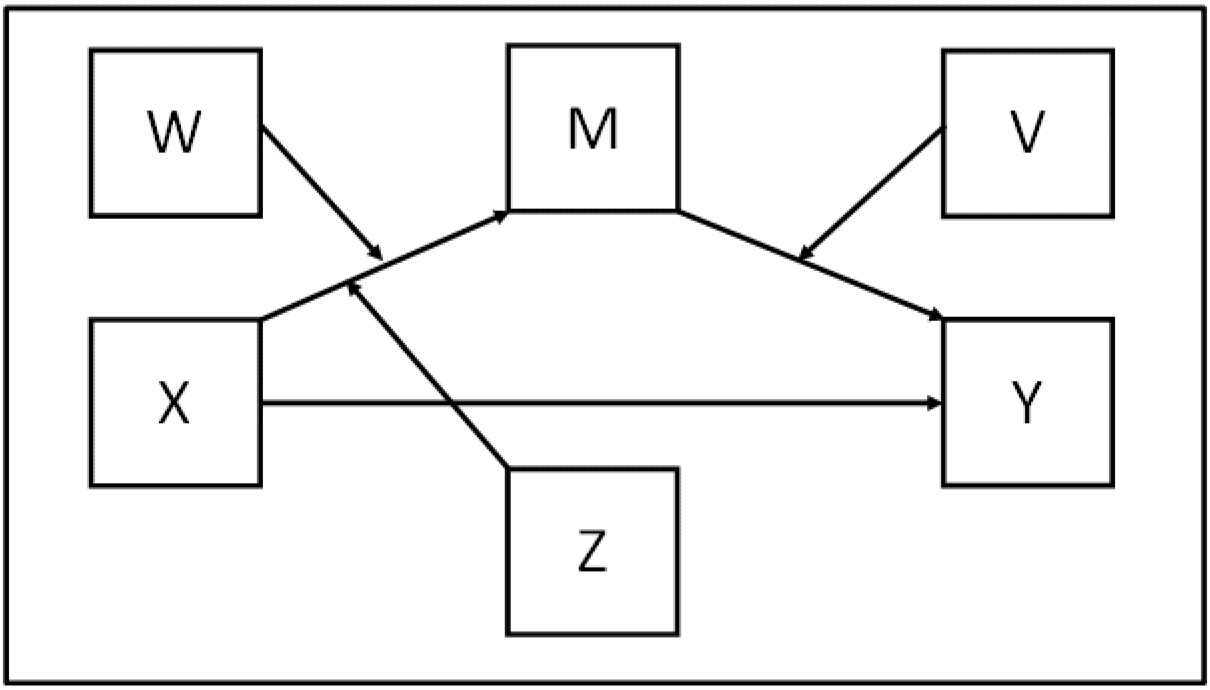
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

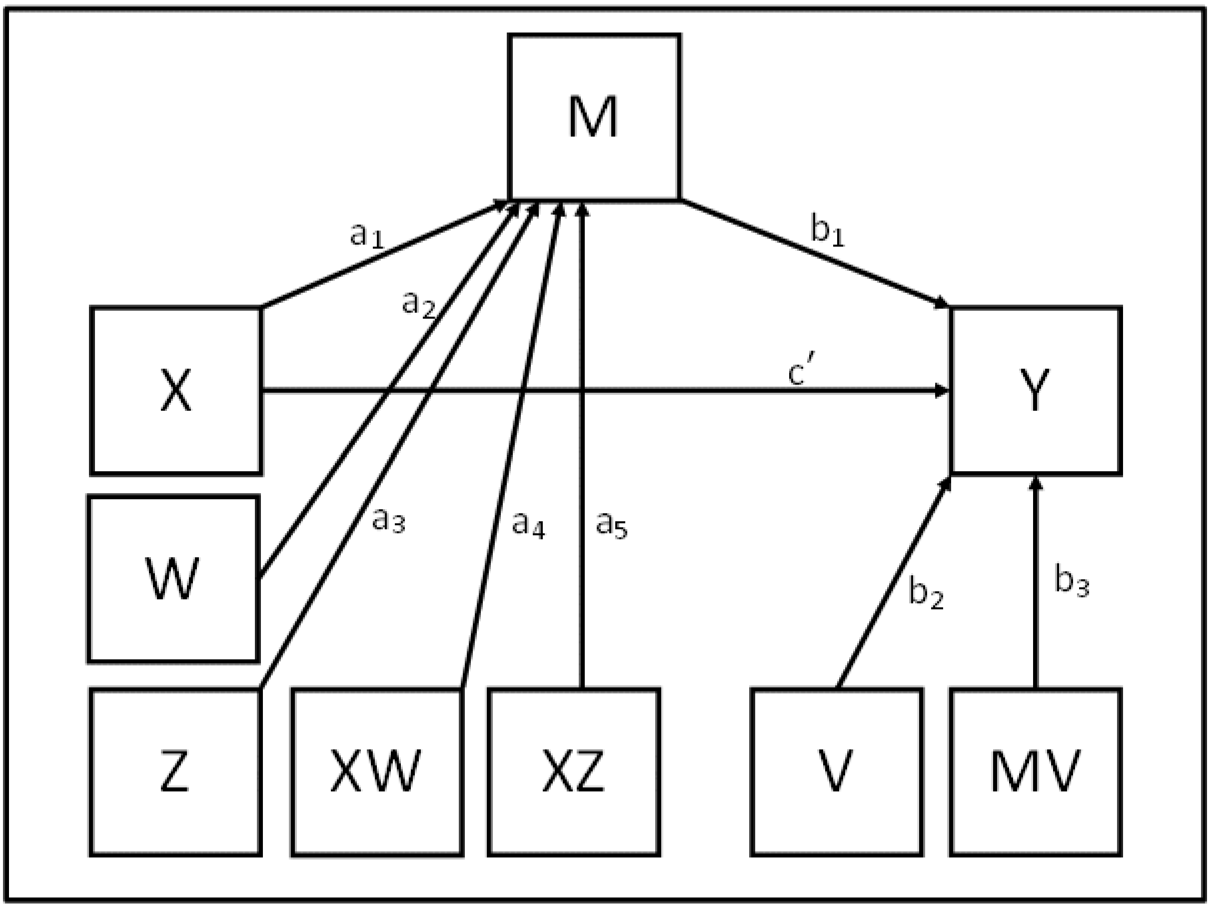
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3MV + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3MV + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2V + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ)V + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + b2V + a0b3V + a1b3XV + a2b3VW + a3b3ZV + a4b3XVW + a5b3XZV + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + b2V + a0b3V + a2b3VW + a3b3ZV) + (a1b1 + a4b1W + a5b1Z + a1b3V + a4b3VW + a5b3ZV + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V:

a1b1 + a4b1W + a5b1Z + a1b3V + a4b3VW + a5b3ZV = (a1 + a4W + a5Z)(b1 + b3V)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON MV (b3);

   Y ON X (cdash);
```

```
  [M] (a0);
  M ON X (a1);
  M ON W (a2);
  M ON Z (a3);
  M ON XW (a4);
  M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
    ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
    ILWHZLV IMWHZLV IHWHZLV
    ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
    ILWHZMV IMWHZMV IHWHZMV
    ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
    ILWHZHV IMWHZHV IHWHZHV
    TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
    TLWHZLV TMWHZLV THWHZLV
    TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
    TLWHZMV TMWHZMV THWHZMV
    TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
    TLWHZHV TMWHZHV THWHZHV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z
```

```
    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*LOW_W + a5*b3*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*MED_W + a5*b3*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*HIGH_W + a5*b3*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*LOW_W + a5*b3*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*MED_W + a5*b3*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*HIGH_W + a5*b3*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*LOW_W + a5*b3*HIGH_Z*LOW_V;
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*MED_W + a5*b3*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b3*LOW_V +
     a4*b3*LOW_V*HIGH_W + a5*b3*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b3*MED_V
+
     a4*b3*MED_V*LOW_W + a5*b3*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b3*MED_V
+
     a4*b3*MED_V*MED_W + a5*b3*LOW_Z*MED_V;
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b3*MED_V
+
     a4*b3*MED_V*HIGH_W + a5*b3*LOW_Z*MED_V;
```

```
    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b3*MED_V
+
     a4*b3*MED_V*LOW_W + a5*b3*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b3*MED_V
+
     a4*b3*MED_V*MED_W + a5*b3*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b3*MED_V
+
     a4*b3*MED_V*HIGH_W + a5*b3*MED_Z*MED_V;

    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b3*MED_V
+
     a4*b3*MED_V*LOW_W + a5*b3*HIGH_Z*MED_V;
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b3*MED_V
+
     a4*b3*MED_V*MED_W + a5*b3*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b3*MED_V +
     a4*b3*MED_V*HIGH_W + a5*b3*HIGH_Z*MED_V;

    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b3*HIGH_V
+
     a4*b3*HIGH_V*LOW_W + a5*b3*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b3*HIGH_V
+
     a4*b3*HIGH_V*MED_W + a5*b3*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a1*b3*HIGH_V +
     a4*b3*HIGH_V*HIGH_W + a5*b3*LOW_Z*HIGH_V;

    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b3*HIGH_V
+
     a4*b3*HIGH_V*LOW_W + a5*b3*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b3*HIGH_V
+
     a4*b3*HIGH_V*MED_W + a5*b3*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b3*HIGH_V +
     a4*b3*HIGH_V*HIGH_W + a5*b3*MED_Z*HIGH_V;

    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b3*HIGH_V +
     a4*b3*HIGH_V*LOW_W + a5*b3*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b3*HIGH_V +
     a4*b3*HIGH_V*MED_W + a5*b3*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b3*HIGH_V +
     a4*b3*HIGH_V*HIGH_W + a5*b3*HIGH_Z*HIGH_V;
```

```
! Calc conditional total effects for each combination of
moderator values

    TLWLZLV = ILWLZLV + cdash;
    TMWLZLV = IMWLZLV + cdash;
    THWLZLV = IHWLZLV + cdash;

    TLWMZLV = ILWMZLV + cdash;
    TMWMZLV = IMWMZLV + cdash;
    THWMZLV = IHWMZLV + cdash;

    TLWHZLV = ILWHZLV + cdash;
    TMWHZLV = IMWHZLV + cdash;
    THWHZLV = IHWHZLV + cdash;

    TLWLZMV = ILWLZMV + cdash;
    TMWLZMV = IMWLZMV + cdash;
    THWLZMV = IHWLZMV + cdash;

    TLWMZMV = ILWMZMV + cdash;
    TMWMZMV = IMWMZMV + cdash;
    THWMZMV = IHWMZMV + cdash;

    TLWHZMV = ILWHZMV + cdash;
    TMWHZMV = IMWHZMV + cdash;
    THWHZMV = IHWHZMV + cdash;

    TLWLZHV = ILWLZHV + cdash;
    TMWLZHV = IMWLZHV + cdash;
    THWLZHV = IHWLZHV + cdash;

    TLWMZHV = ILWMZHV + cdash;
    TMWMZHV = IMWMZHV + cdash;
    THWMZHV = IHWMZHV + cdash;

    TLWHZHV = ILWHZHV + cdash;
    TMWHZHV = IMWHZHV + cdash;
    THWHZHV = IHWHZHV + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
```

```
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);

    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;

    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;

    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;

    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 24: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating both the IV- Mediator path and direct IV-DV path, 1 moderating the Mediator-DV path

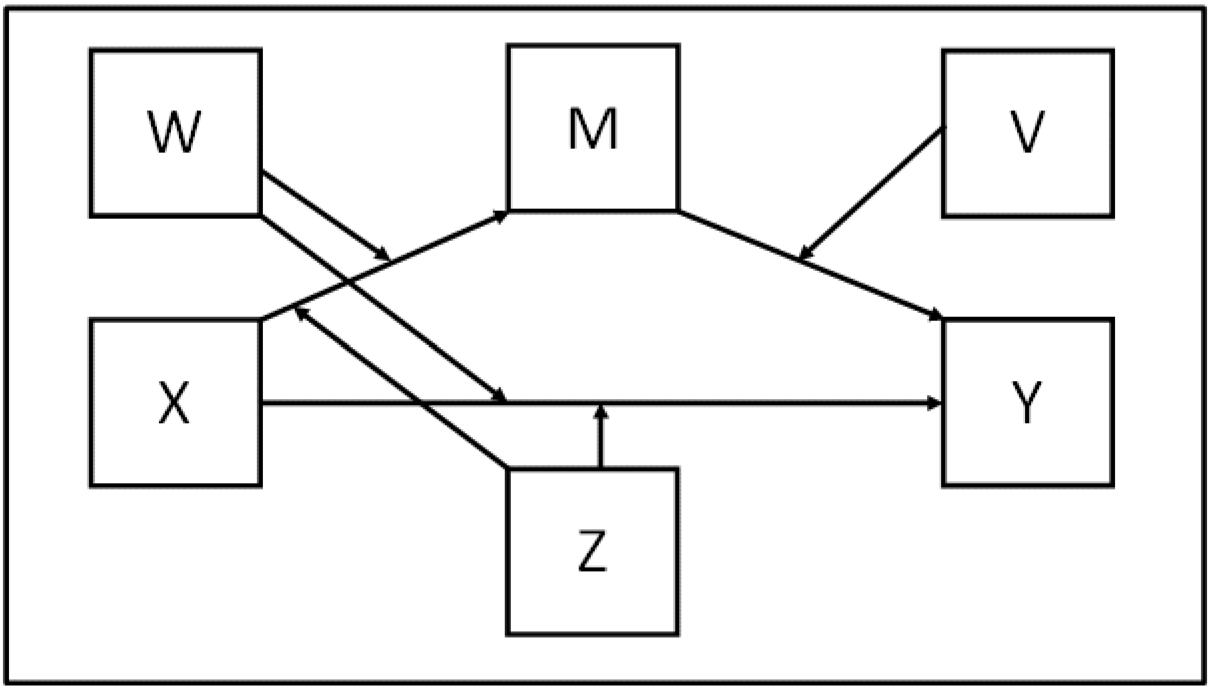
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

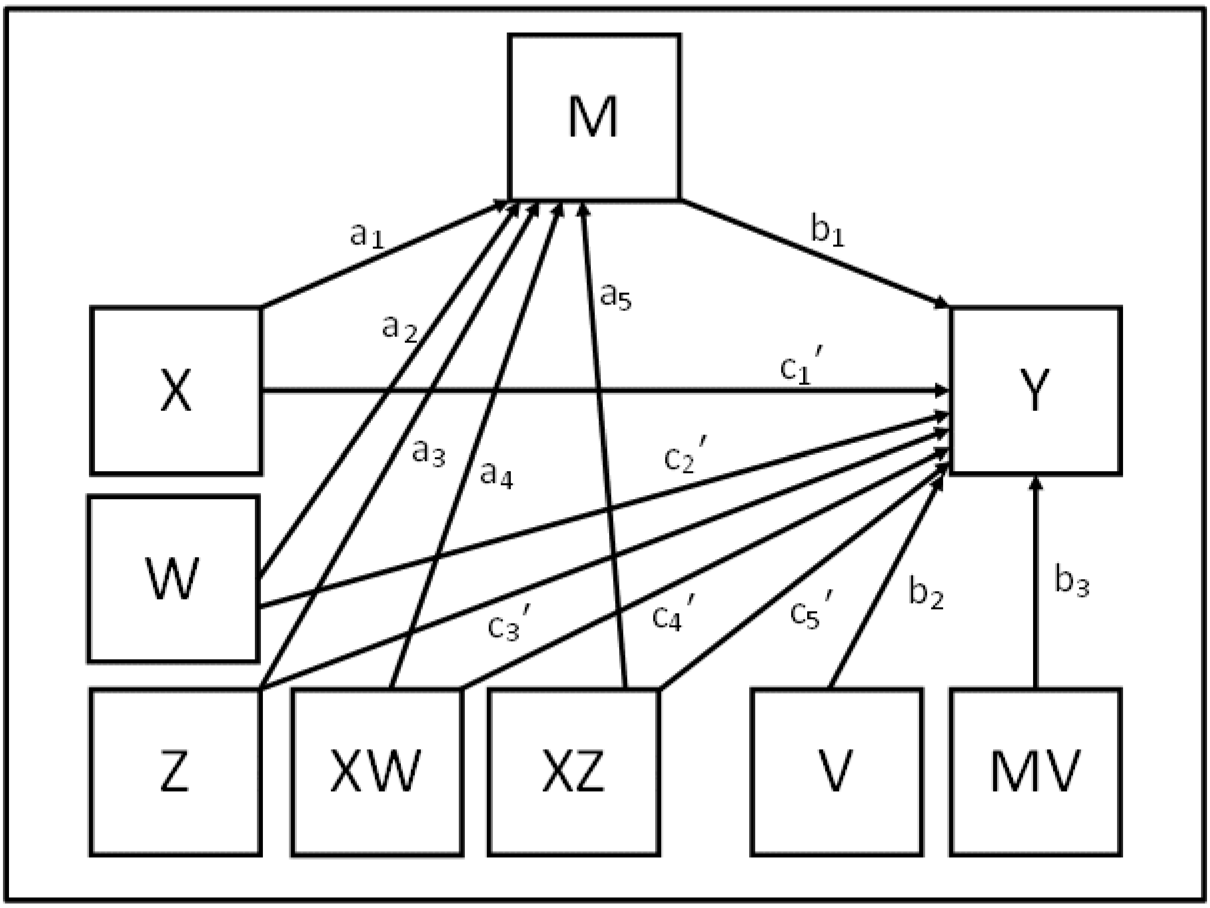
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3MV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3MV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2V + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ)V + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + b2V + a0b3V + a1b3XV + a2b3VW + a3b3ZV + a4b3XVW + a5b3XZV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + b2V + a0b3V + a2b3VW + a3b3ZV + c2'W + c3'Z) + (a1b1 + a4b1W + a5b1Z + a1b3V + a4b3VW + a5b3ZV + c1' + c4'W + c5'Z)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V:

a1b1 + a4b1W + a5b1Z + a1b3V + a4b3VW + a5b3ZV = (a1 + a4W + a5Z)(b1 + b3V)

One direct effect of X on Y, conditional on W, Z:

c1' + c4'W + c5'Z

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON MV (b3);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON Z (cdash3);
```

```
   Y ON XW (cdash4);
   Y ON XZ (cdash5);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
    ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
    ILWHZLV IMWHZLV IHWHZLV
    ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
    ILWHZMV IMWHZMV IHWHZMV
    ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
    ILWHZHV IMWHZHV IHWHZHV
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
    TLWHZLV TMWHZLV THWHZLV
    TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
    TLWHZMV TMWHZMV THWHZMV
    TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
    TLWHZHV TMWHZHV THWHZHV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
```

```
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*LOW_W + a5*b3*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*MED_W + a5*b3*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*HIGH_W + a5*b3*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*LOW_W + a5*b3*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*MED_W + a5*b3*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*HIGH_W + a5*b3*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*LOW_W + a5*b3*HIGH_Z*LOW_V;
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b3*LOW_V
+
     a4*b3*LOW_V*MED_W + a5*b3*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b3*LOW_V +
     a4*b3*LOW_V*HIGH_W + a5*b3*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b3*MED_V
+
     a4*b3*MED_V*LOW_W + a5*b3*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b3*MED_V
+
     a4*b3*MED_V*MED_W + a5*b3*LOW_Z*MED_V;
```

```
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b3*MED_V
+
     a4*b3*MED_V*HIGH_W + a5*b3*LOW_Z*MED_V;
    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b3*MED_V
+
     a4*b3*MED_V*LOW_W + a5*b3*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b3*MED_V
+
     a4*b3*MED_V*MED_W + a5*b3*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b3*MED_V
+
     a4*b3*MED_V*HIGH_W + a5*b3*MED_Z*MED_V;
    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b3*MED_V
+
     a4*b3*MED_V*LOW_W + a5*b3*HIGH_Z*MED_V;
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b3*MED_V
+
     a4*b3*MED_V*MED_W + a5*b3*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b3*MED_V +
     a4*b3*MED_V*HIGH_W + a5*b3*HIGH_Z*MED_V;
    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b3*HIGH_V
+
     a4*b3*HIGH_V*LOW_W + a5*b3*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b3*HIGH_V
+
     a4*b3*HIGH_V*MED_W + a5*b3*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a1*b3*HIGH_V +
     a4*b3*HIGH_V*HIGH_W + a5*b3*LOW_Z*HIGH_V;
    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b3*HIGH_V
+
     a4*b3*HIGH_V*LOW_W + a5*b3*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b3*HIGH_V
+
     a4*b3*HIGH_V*MED_W + a5*b3*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b3*HIGH_V +
     a4*b3*HIGH_V*HIGH_W + a5*b3*MED_Z*HIGH_V;
    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b3*HIGH_V +
     a4*b3*HIGH_V*LOW_W + a5*b3*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b3*HIGH_V +
```

```
     a4*b3*HIGH_V*MED_W + a5*b3*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b3*HIGH_V +
     a4*b3*HIGH_V*HIGH_W + a5*b3*HIGH_Z*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLWLZLV = ILWLZLV + DLOW_LOZ;
    TMWLZLV = IMWLZLV + DMEW_LOZ;
    THWLZLV = IHWLZLV + DHIW_LOZ;

    TLWMZLV = ILWMZLV + DLOW_MEZ;
    TMWMZLV = IMWMZLV + DMEW_MEZ;
    THWMZLV = IHWMZLV + DHIW_MEZ;

    TLWHZLV = ILWHZLV + DLOW_HIZ;
    TMWHZLV = IMWHZLV + DMEW_HIZ;
    THWHZLV = IHWHZLV + DHIW_HIZ;

    TLWLZMV = ILWLZMV + DLOW_LOZ;
    TMWLZMV = IMWLZMV + DMEW_LOZ;
    THWLZMV = IHWLZMV + DHIW_LOZ;

    TLWMZMV = ILWMZMV + DLOW_MEZ;
    TMWMZMV = IMWMZMV + DMEW_MEZ;
    THWMZMV = IHWMZMV + DHIW_MEZ;

    TLWHZMV = ILWHZMV + DLOW_HIZ;
    TMWHZMV = IMWHZMV + DMEW_HIZ;
    THWHZMV = IHWHZMV + DHIW_HIZ;

    TLWLZHV = ILWLZHV + DLOW_LOZ;
    TMWLZHV = IMWLZHV + DMEW_LOZ;
    THWLZHV = IHWLZHV + DHIW_LOZ;
```

```
    TLWMZHV = ILWMZHV + DLOW_MEZ;
    TMWMZHV = IMWMZHV + DMEW_MEZ;
    THWMZHV = IHWMZHV + DHIW_MEZ;

    TLWHZHV = ILWHZHV + DLOW_HIZ;
    TMWHZHV = IMWHZHV + DMEW_HIZ;
    THWHZHV = IHWHZHV + DHIW_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);

    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;

    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;

    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;
```

```
    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 25: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path with all 2-way and 3-way interactions, 1 moderating the Mediator-DV path

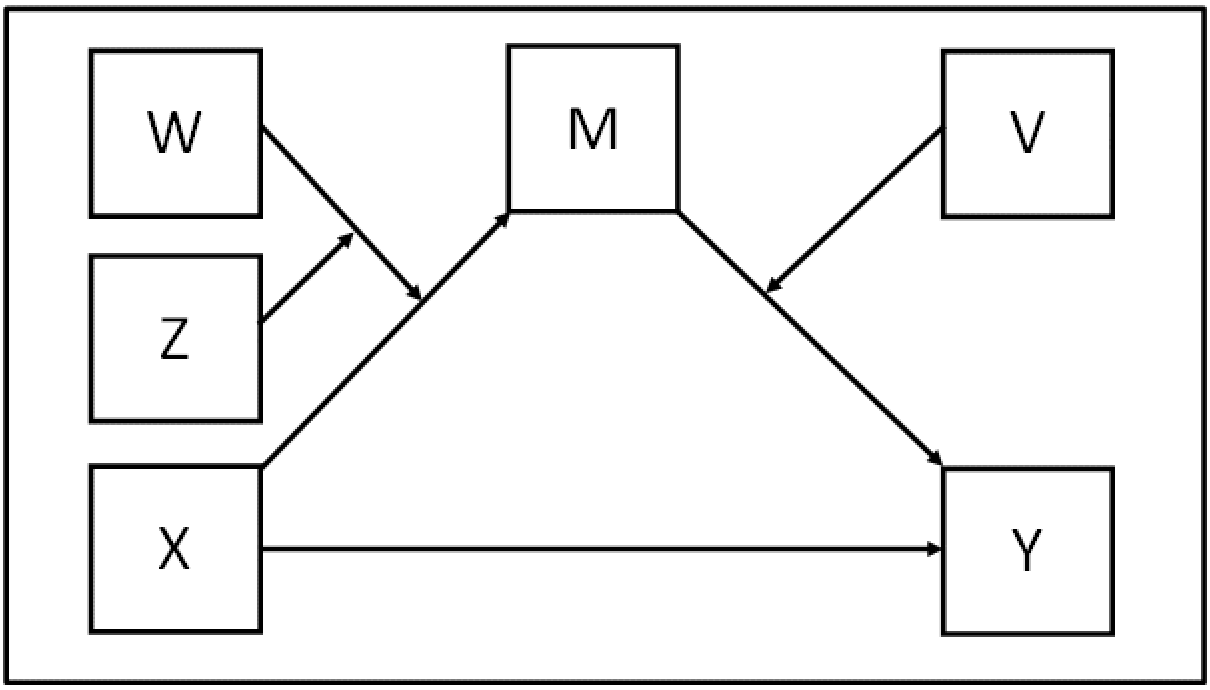
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

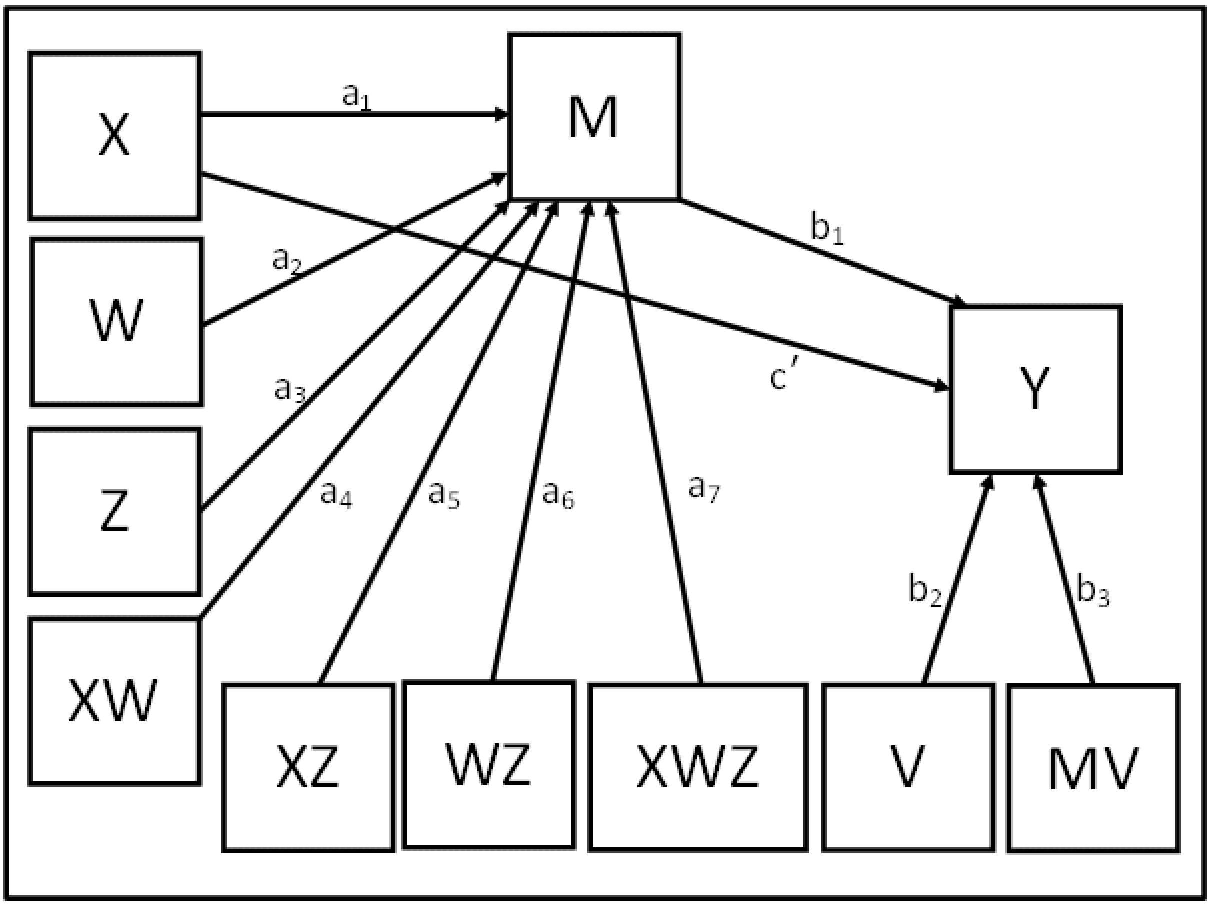
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3MV + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3MV + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2V + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + b2V + a0b3V + a1b3XV + a2b3WV + a3b3ZV + a4b3XWV + a5b3XZV + a6b3WZV + a7b3XWZV + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + b2V + a0b3V + a2b3WV + a3b3ZV + a6b3WZV) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b3V + a4b3WV + a5b3ZV + a7b3WZV + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b3V + a4b3WV + a5b3ZV + a7b3WZV = (a1 + a4W + a5Z + a7WZ)(b1 + b3V)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ WZ MV XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON V (b2);
    Y ON MV (b3);

    Y ON X (cdash);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
    M ON WZ (a6);
    M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
     NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
     ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
     ILWHZLV IMWHZLV IHWHZLV
     ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
     ILWHZMV IMWHZMV IHWHZMV
     ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
     ILWHZHV IMWHZHV IHWHZHV
     TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
     TLWHZLV TMWHZLV THWHZLV
     TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
     TLWHZMV TMWHZMV THWHZMV
     TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
     TLWHZHV TMWHZHV THWHZHV);

     LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
     MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
```

```
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*LOW_V + a4*b3*LOW_W*LOW_V + a5*b3*LOW_Z*LOW_V +
     a7*b3*LOW_W*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*LOW_V + a4*b3*MED_W*LOW_V + a5*b3*LOW_Z*LOW_V +
     a7*b3*MED_W*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*LOW_V + a4*b3*HIGH_W*LOW_V + a5*b3*LOW_Z*LOW_V +
     a7*b3*HIGH_W*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*LOW_V + a4*b3*LOW_W*LOW_V + a5*b3*MED_Z*LOW_V +
     a7*b3*LOW_W*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*LOW_V + a4*b3*MED_W*LOW_V + a5*b3*MED_Z*LOW_V +
     a7*b3*MED_W*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*LOW_V + a4*b3*HIGH_W*LOW_V + a5*b3*MED_Z*LOW_V +
     a7*b3*HIGH_W*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b3*LOW_V + a4*b3*LOW_W*LOW_V + a5*b3*HIGH_Z*LOW_V +
     a7*b3*LOW_W*HIGH_Z*LOW_V;
```

```
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*LOW_V + a4*b3*MED_W*LOW_V + a5*b3*HIGH_Z*LOW_V +
     a7*b3*MED_W*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b3*LOW_V + a4*b3*HIGH_W*LOW_V + a5*b3*HIGH_Z*LOW_V +
     a7*b3*HIGH_W*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*MED_V + a4*b3*LOW_W*MED_V + a5*b3*LOW_Z*MED_V +
     a7*b3*LOW_W*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*MED_V + a4*b3*MED_W*MED_V + a5*b3*LOW_Z*MED_V +
     a7*b3*MED_W*LOW_Z*MED_V;
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*MED_V + a4*b3*HIGH_W*MED_V + a5*b3*LOW_Z*MED_V +
     a7*b3*HIGH_W*LOW_Z*MED_V;

    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*MED_V + a4*b3*LOW_W*MED_V + a5*b3*MED_Z*MED_V +
     a7*b3*LOW_W*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*MED_V + a4*b3*MED_W*MED_V + a5*b3*MED_Z*MED_V +
     a7*b3*MED_W*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*MED_V + a4*b3*HIGH_W*MED_V + a5*b3*MED_Z*MED_V +
     a7*b3*HIGH_W*MED_Z*MED_V;

    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b3*MED_V + a4*b3*LOW_W*MED_V + a5*b3*HIGH_Z*MED_V +
     a7*b3*LOW_W*HIGH_Z*MED_V;
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*MED_V + a4*b3*MED_W*MED_V + a5*b3*HIGH_Z*MED_V +
     a7*b3*MED_W*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b3*MED_V + a4*b3*HIGH_W*MED_V + a5*b3*HIGH_Z*MED_V +
     a7*b3*HIGH_W*HIGH_Z*MED_V;
```

```
    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*HIGH_V + a4*b3*LOW_W*HIGH_V + a5*b3*LOW_Z*HIGH_V +
     a7*b3*LOW_W*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*HIGH_V + a4*b3*MED_W*HIGH_V + a5*b3*LOW_Z*HIGH_V +
     a7*b3*MED_W*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*HIGH_V + a4*b3*HIGH_W*HIGH_V + a5*b3*LOW_Z*HIGH_V
+
     a7*b3*HIGH_W*LOW_Z*HIGH_V;

    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*HIGH_V + a4*b3*LOW_W*HIGH_V + a5*b3*MED_Z*HIGH_V +
     a7*b3*LOW_W*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*HIGH_V + a4*b3*MED_W*HIGH_V + a5*b3*MED_Z*HIGH_V +
     a7*b3*MED_W*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*HIGH_V + a4*b3*HIGH_W*HIGH_V + a5*b3*MED_Z*HIGH_V
+
     a7*b3*HIGH_W*MED_Z*HIGH_V;

    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b3*HIGH_V + a4*b3*LOW_W*HIGH_V + a5*b3*HIGH_Z*HIGH_V
+
     a7*b3*LOW_W*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*HIGH_V + a4*b3*MED_W*HIGH_V + a5*b3*HIGH_Z*HIGH_V
+
     a7*b3*MED_W*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b3*HIGH_V + a4*b3*HIGH_W*HIGH_V + a5*b3*HIGH_Z*HIGH_V
+
     a7*b3*HIGH_W*HIGH_Z*HIGH_V;

! Calc conditional total effects for each combination of
moderator values
```

```
    TLWLZLV = ILWLZLV + cdash;
    TMWLZLV = IMWLZLV + cdash;
    THWLZLV = IHWLZLV + cdash;

    TLWMZLV = ILWMZLV + cdash;
    TMWMZLV = IMWMZLV + cdash;
    THWMZLV = IHWMZLV + cdash;

    TLWHZLV = ILWHZLV + cdash;
    TMWHZLV = IMWHZLV + cdash;
    THWHZLV = IHWHZLV + cdash;

    TLWLZMV = ILWLZMV + cdash;
    TMWLZMV = IMWLZMV + cdash;
    THWLZMV = IHWLZMV + cdash;

    TLWMZMV = ILWMZMV + cdash;
    TMWMZMV = IMWMZMV + cdash;
    THWMZMV = IHWMZMV + cdash;

    TLWHZMV = ILWHZMV + cdash;
    TMWHZMV = IMWHZMV + cdash;
    THWHZMV = IHWHZMV + cdash;

    TLWLZHV = ILWLZHV + cdash;
    TMWLZHV = IMWLZHV + cdash;
    THWLZHV = IHWLZHV + cdash;

    TLWMZHV = ILWMZHV + cdash;
    TMWMZHV = IMWMZHV + cdash;
    THWMZHV = IHWMZHV + cdash;

    TLWHZHV = ILWHZHV + cdash;
    TMWHZHV = IMWHZHV + cdash;
    THWHZHV = IHWHZHV + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);
```

```
    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;

    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;

    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;

    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 26: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating both the IV- Mediator path and direct IV-DV path with all 2-way and 3-way interactions, 1 moderating the Mediator-DV path

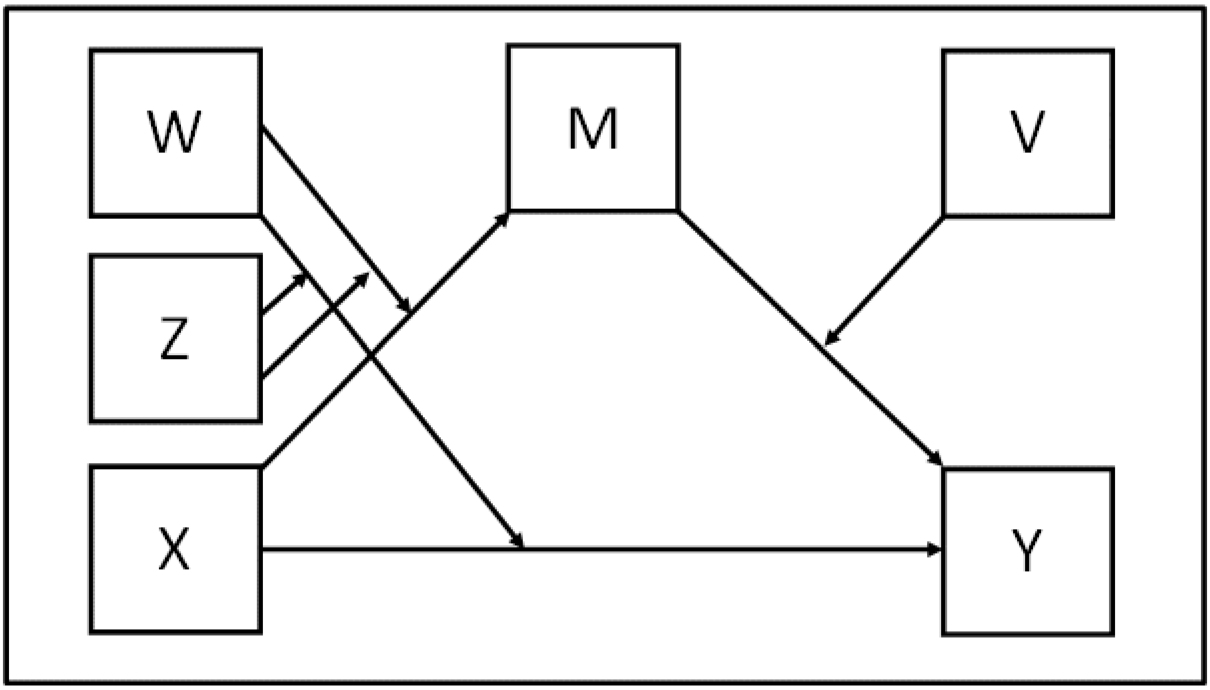
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

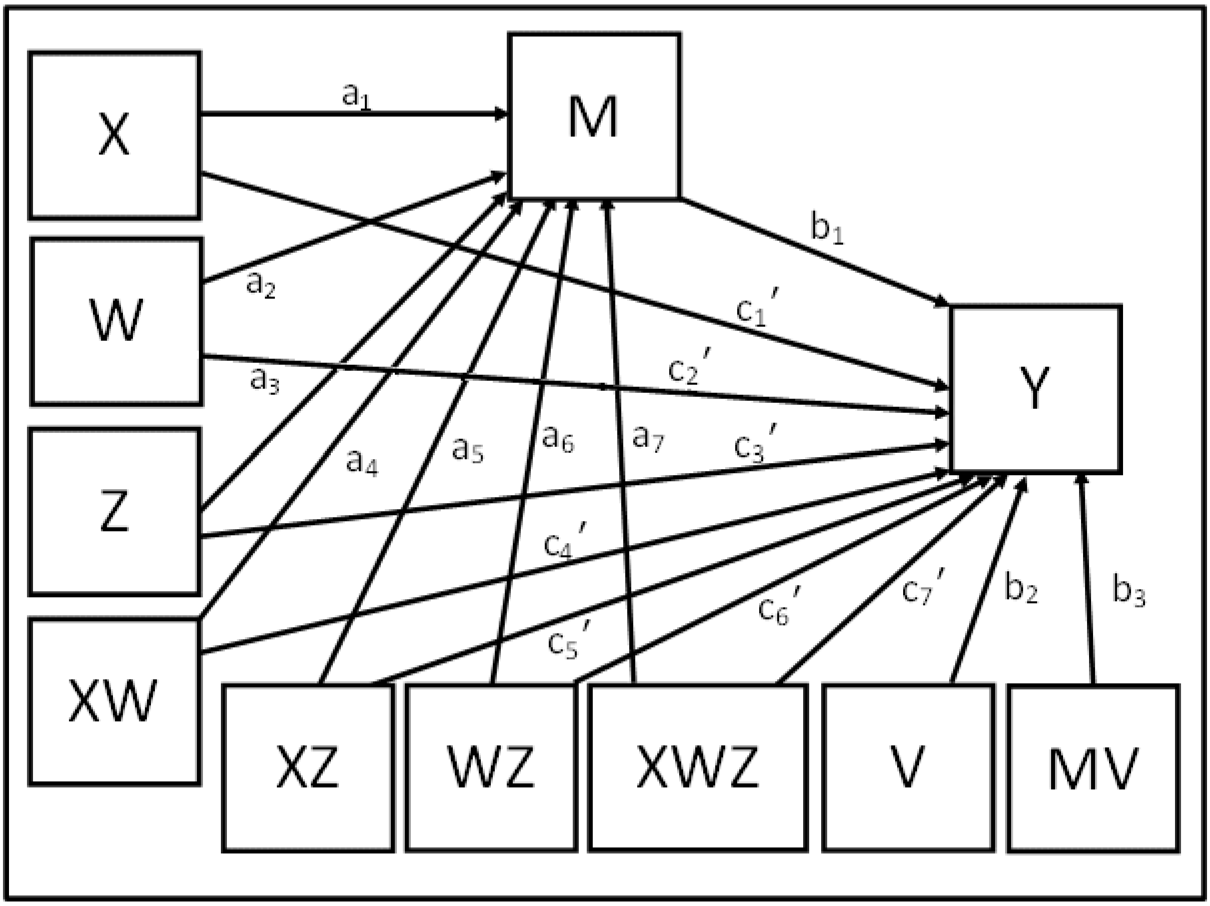
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3MV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3MV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2V + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + b2V + a0b3 + a1b3XV + a2b3WV + a3b3ZV + a4b3XWV + a5b3XZV + a6b3WZV + a7b3XWZV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + a7b1XWZ + b2V + a0b3 + a2b3WV + a3b3ZV + a6b3WZV + c2'W + c3'Z + c6'WZ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b3V + a4b3WV + a5b3ZV + a7b3WZV + c1' + c4'W + c5'Z + c7'WZ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b3V + a4b3WV + a5b3ZV + a7b3WZV = (a1 + a4W + a5Z + a7WZ)(b1 + b3V)

One direct effect of X on Y, conditional on W, Z:

c1' + c4'W + c5'Z + c7'WZ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ WZ MV XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;
```

```
! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON MV (b3);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON Z (cdash3);
   Y ON XW (cdash4);
   Y ON XZ (cdash5);
   Y ON WZ (cdash6);
   Y ON XWZ (cdash7);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
    ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
    ILWHZLV IMWHZLV IHWHZLV
    ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
    ILWHZMV IMWHZMV IHWHZMV
    ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
    ILWHZHV IMWHZHV IHWHZHV
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
```

```
    TLWHZLV TMWHZLV THWHZLV
    TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
    TLWHZMV TMWHZMV THWHZMV
    TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
    TLWHZHV TMWHZHV THWHZHV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*LOW_V + a4*b3*LOW_W*LOW_V + a5*b3*LOW_Z*LOW_V +
     a7*b3*LOW_W*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*LOW_V + a4*b3*MED_W*LOW_V + a5*b3*LOW_Z*LOW_V +
     a7*b3*MED_W*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*LOW_V + a4*b3*HIGH_W*LOW_V + a5*b3*LOW_Z*LOW_V +
     a7*b3*HIGH_W*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*LOW_V + a4*b3*LOW_W*LOW_V + a5*b3*MED_Z*LOW_V +
     a7*b3*LOW_W*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*LOW_V + a4*b3*MED_W*LOW_V + a5*b3*MED_Z*LOW_V +
```

```
     a7*b3*MED_W*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*LOW_V + a4*b3*HIGH_W*LOW_V + a5*b3*MED_Z*LOW_V +
     a7*b3*HIGH_W*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b3*LOW_V + a4*b3*LOW_W*LOW_V + a5*b3*HIGH_Z*LOW_V +
     a7*b3*LOW_W*HIGH_Z*LOW_V;
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*LOW_V + a4*b3*MED_W*LOW_V + a5*b3*HIGH_Z*LOW_V +
     a7*b3*MED_W*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b3*LOW_V + a4*b3*HIGH_W*LOW_V + a5*b3*HIGH_Z*LOW_V +
     a7*b3*HIGH_W*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*MED_V + a4*b3*LOW_W*MED_V + a5*b3*LOW_Z*MED_V +
     a7*b3*LOW_W*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*MED_V + a4*b3*MED_W*MED_V + a5*b3*LOW_Z*MED_V +
     a7*b3*MED_W*LOW_Z*MED_V;
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*MED_V + a4*b3*HIGH_W*MED_V + a5*b3*LOW_Z*MED_V +
     a7*b3*HIGH_W*LOW_Z*MED_V;

    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*MED_V + a4*b3*LOW_W*MED_V + a5*b3*MED_Z*MED_V +
     a7*b3*LOW_W*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*MED_V + a4*b3*MED_W*MED_V + a5*b3*MED_Z*MED_V +
     a7*b3*MED_W*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*MED_V + a4*b3*HIGH_W*MED_V + a5*b3*MED_Z*MED_V +
     a7*b3*HIGH_W*MED_Z*MED_V;

    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b3*MED_V + a4*b3*LOW_W*MED_V + a5*b3*HIGH_Z*MED_V +
     a7*b3*LOW_W*HIGH_Z*MED_V;
```

```
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*MED_V + a4*b3*MED_W*MED_V + a5*b3*HIGH_Z*MED_V +
     a7*b3*MED_W*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b3*MED_V + a4*b3*HIGH_W*MED_V + a5*b3*HIGH_Z*MED_V +
     a7*b3*HIGH_W*HIGH_Z*MED_V;

    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*HIGH_V + a4*b3*LOW_W*HIGH_V + a5*b3*LOW_Z*HIGH_V +
     a7*b3*LOW_W*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*HIGH_V + a4*b3*MED_W*HIGH_V + a5*b3*LOW_Z*HIGH_V +
     a7*b3*MED_W*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*HIGH_V + a4*b3*HIGH_W*HIGH_V + a5*b3*LOW_Z*HIGH_V
+
     a7*b3*HIGH_W*LOW_Z*HIGH_V;

    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*HIGH_V + a4*b3*LOW_W*HIGH_V + a5*b3*MED_Z*HIGH_V +
     a7*b3*LOW_W*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*HIGH_V + a4*b3*MED_W*HIGH_V + a5*b3*MED_Z*HIGH_V +
     a7*b3*MED_W*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*HIGH_V + a4*b3*HIGH_W*HIGH_V + a5*b3*MED_Z*HIGH_V
+
     a7*b3*HIGH_W*MED_Z*HIGH_V;

    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b3*HIGH_V + a4*b3*LOW_W*HIGH_V + a5*b3*HIGH_Z*HIGH_V
+
     a7*b3*LOW_W*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*HIGH_V + a4*b3*MED_W*HIGH_V + a5*b3*HIGH_Z*HIGH_V
+
     a7*b3*MED_W*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
```

```
     a1*b3*HIGH_V + a4*b3*HIGH_W*HIGH_V + a5*b3*HIGH_Z*HIGH_V
+
     a7*b3*HIGH_W*HIGH_Z*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLWLZLV = ILWLZLV + DLOW_LOZ;
    TMWLZLV = IMWLZLV + DMEW_LOZ;
    THWLZLV = IHWLZLV + DHIW_LOZ;

    TLWMZLV = ILWMZLV + DLOW_MEZ;
    TMWMZLV = IMWMZLV + DMEW_MEZ;
    THWMZLV = IHWMZLV + DHIW_MEZ;

    TLWHZLV = ILWHZLV + DLOW_HIZ;
    TMWHZLV = IMWHZLV + DMEW_HIZ;
    THWHZLV = IHWHZLV + DHIW_HIZ;

    TLWLZMV = ILWLZMV + DLOW_LOZ;
    TMWLZMV = IMWLZMV + DMEW_LOZ;
    THWLZMV = IHWLZMV + DHIW_LOZ;

    TLWMZMV = ILWMZMV + DLOW_MEZ;
    TMWMZMV = IMWMZMV + DMEW_MEZ;
    THWMZMV = IHWMZMV + DHIW_MEZ;
```

```
    TLWHZMV = ILWHZMV + DLOW_HIZ;
    TMWHZMV = IMWHZMV + DMEW_HIZ;
    THWHZMV = IHWHZMV + DHIW_HIZ;

    TLWLZHV = ILWLZHV + DLOW_LOZ;
    TMWLZHV = IMWLZHV + DMEW_LOZ;
    THWLZHV = IHWLZHV + DHIW_LOZ;

    TLWMZHV = ILWMZHV + DLOW_MEZ;
    TMWMZHV = IMWMZHV + DMEW_MEZ;
    THWMZHV = IHWMZHV + DHIW_MEZ;

    TLWHZHV = ILWHZHV + DLOW_HIZ;
    TMWHZHV = IMWHZHV + DMEW_HIZ;
    THWHZHV = IHWHZHV + DHIW_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);

    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;

    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;
```

```
    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;

    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 27: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path with all 2-way and 3-way interactions, one of which also moderates the direct IV-DV path, 1 moderating the Mediator-DV path

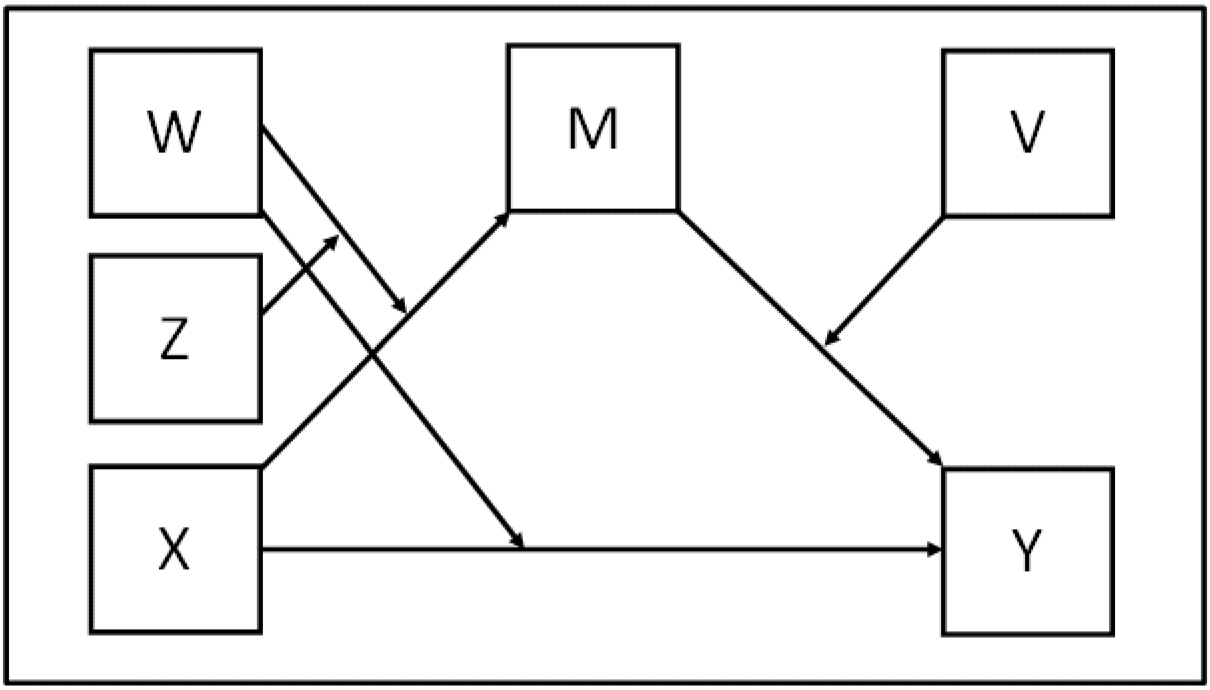
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

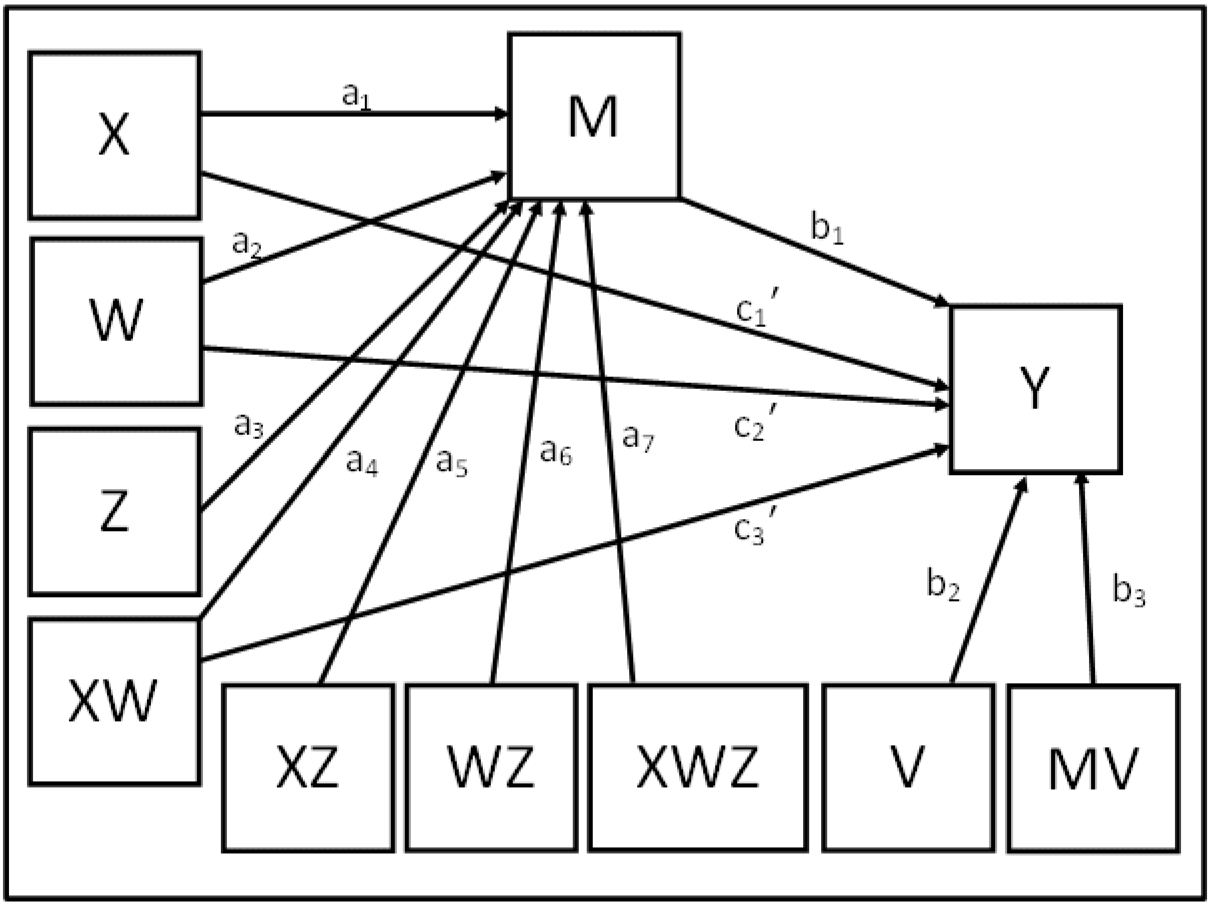
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3MV + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3MV + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2V + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + b2V + a0b3V + a1b3XV + a2b3WV + a3b3ZV + a4b3XWV + a5b3XZV + a6b3WZV + a7b3XWZV + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + b2V + a0b3V + a2b3WV + a3b3ZV + a6b3WZV + c2'W) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b3V + a4b3WV + a5b3ZV + a7b3WZV + c1' + c3'W)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b3V + a4b3WV + a5b3ZV + a7b3WZV = (a1 + a4W + a5Z + a7WZ)(b1 + b3V)

One direct effect of X on Y, conditional on W:

c1' + c3'W

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ WZ MV XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON MV (b3);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
    ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
    ILWHZLV IMWHZLV IHWHZLV
    ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
    ILWHZMV IMWHZMV IHWHZMV
    ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
    ILWHZHV IMWHZHV IHWHZHV
    DIR_LOWW DIR_MEDW DIR_HIW
    TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
    TLWHZLV TMWHZLV THWHZLV
    TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
    TLWHZMV TMWHZMV THWHZMV
    TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
    TLWHZHV TMWHZHV THWHZHV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
```

```
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*LOW_V + a4*b3*LOW_W*LOW_V + a5*b3*LOW_Z*LOW_V +
     a7*b3*LOW_W*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*LOW_V + a4*b3*MED_W*LOW_V + a5*b3*LOW_Z*LOW_V +
     a7*b3*MED_W*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*LOW_V + a4*b3*HIGH_W*LOW_V + a5*b3*LOW_Z*LOW_V +
     a7*b3*HIGH_W*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*LOW_V + a4*b3*LOW_W*LOW_V + a5*b3*MED_Z*LOW_V +
     a7*b3*LOW_W*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*LOW_V + a4*b3*MED_W*LOW_V + a5*b3*MED_Z*LOW_V +
     a7*b3*MED_W*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*LOW_V + a4*b3*HIGH_W*LOW_V + a5*b3*MED_Z*LOW_V +
     a7*b3*HIGH_W*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
```

```
     a1*b3*LOW_V + a4*b3*LOW_W*LOW_V + a5*b3*HIGH_Z*LOW_V +
     a7*b3*LOW_W*HIGH_Z*LOW_V;
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*LOW_V + a4*b3*MED_W*LOW_V + a5*b3*HIGH_Z*LOW_V +
     a7*b3*MED_W*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b3*LOW_V + a4*b3*HIGH_W*LOW_V + a5*b3*HIGH_Z*LOW_V +
     a7*b3*HIGH_W*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*MED_V + a4*b3*LOW_W*MED_V + a5*b3*LOW_Z*MED_V +
     a7*b3*LOW_W*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*MED_V + a4*b3*MED_W*MED_V + a5*b3*LOW_Z*MED_V +
     a7*b3*MED_W*LOW_Z*MED_V;
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*MED_V + a4*b3*HIGH_W*MED_V + a5*b3*LOW_Z*MED_V +
     a7*b3*HIGH_W*LOW_Z*MED_V;

    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*MED_V + a4*b3*LOW_W*MED_V + a5*b3*MED_Z*MED_V +
     a7*b3*LOW_W*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*MED_V + a4*b3*MED_W*MED_V + a5*b3*MED_Z*MED_V +
     a7*b3*MED_W*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*MED_V + a4*b3*HIGH_W*MED_V + a5*b3*MED_Z*MED_V +
     a7*b3*HIGH_W*MED_Z*MED_V;

    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b3*MED_V + a4*b3*LOW_W*MED_V + a5*b3*HIGH_Z*MED_V +
     a7*b3*LOW_W*HIGH_Z*MED_V;
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*MED_V + a4*b3*MED_W*MED_V + a5*b3*HIGH_Z*MED_V +
     a7*b3*MED_W*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b3*MED_V + a4*b3*HIGH_W*MED_V + a5*b3*HIGH_Z*MED_V +
     a7*b3*HIGH_W*HIGH_Z*MED_V;
```

```
    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*HIGH_V + a4*b3*LOW_W*HIGH_V + a5*b3*LOW_Z*HIGH_V +
     a7*b3*LOW_W*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*HIGH_V + a4*b3*MED_W*HIGH_V + a5*b3*LOW_Z*HIGH_V +
     a7*b3*MED_W*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*HIGH_V + a4*b3*HIGH_W*HIGH_V + a5*b3*LOW_Z*HIGH_V
+
     a7*b3*HIGH_W*LOW_Z*HIGH_V;

    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*HIGH_V + a4*b3*LOW_W*HIGH_V + a5*b3*MED_Z*HIGH_V +
     a7*b3*LOW_W*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*HIGH_V + a4*b3*MED_W*HIGH_V + a5*b3*MED_Z*HIGH_V +
     a7*b3*MED_W*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*HIGH_V + a4*b3*HIGH_W*HIGH_V + a5*b3*MED_Z*HIGH_V
+
     a7*b3*HIGH_W*MED_Z*HIGH_V;

    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b3*HIGH_V + a4*b3*LOW_W*HIGH_V + a5*b3*HIGH_Z*HIGH_V
+
     a7*b3*LOW_W*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*HIGH_V + a4*b3*MED_W*HIGH_V + a5*b3*HIGH_Z*HIGH_V
+
     a7*b3*MED_W*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b3*HIGH_V + a4*b3*HIGH_W*HIGH_V + a5*b3*HIGH_Z*HIGH_V
+
     a7*b3*HIGH_W*HIGH_Z*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values
```

```
    DIR_LOWW = cdash1 + cdash3*LOW_W;
    DIR_MEDW = cdash1 + cdash3*MED_W;
    DIR_HIW = cdash1 + cdash3*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TLWLZLV = ILWLZLV + DIR_LOWW;
    TMWLZLV = IMWLZLV + DIR_MEDW;
    THWLZLV = IHWLZLV + DIR_HIW;

    TLWMZLV = ILWMZLV + DIR_LOWW;
    TMWMZLV = IMWMZLV + DIR_MEDW;
    THWMZLV = IHWMZLV + DIR_HIW;

    TLWHZLV = ILWHZLV + DIR_LOWW;
    TMWHZLV = IMWHZLV + DIR_MEDW;
    THWHZLV = IHWHZLV + DIR_HIW;

    TLWLZMV = ILWLZMV + DIR_LOWW;
    TMWLZMV = IMWLZMV + DIR_MEDW;
    THWLZMV = IHWLZMV + DIR_HIW;

    TLWMZMV = ILWMZMV + DIR_LOWW;
    TMWMZMV = IMWMZMV + DIR_MEDW;
    THWMZMV = IHWMZMV + DIR_HIW;

    TLWHZMV = ILWHZMV + DIR_LOWW;
    TMWHZMV = IMWHZMV + DIR_MEDW;
    THWHZMV = IHWHZMV + DIR_HIW;

    TLWLZHV = ILWLZHV + DIR_LOWW;
    TMWLZHV = IMWLZHV + DIR_MEDW;
    THWLZHV = IHWLZHV + DIR_HIW;

    TLWMZHV = ILWMZHV + DIR_LOWW;
    TMWMZHV = IMWMZHV + DIR_MEDW;
    THWMZHV = IHWMZHV + DIR_HIW;

    TLWHZHV = ILWHZHV + DIR_LOWW;
    TMWHZHV = IMWHZHV + DIR_MEDW;
    THWHZHV = IHWHZHV + DIR_HIW;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
```

```
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);

    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;

    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;

    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;

    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 28: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, 1 moderating the IV-Mediator path, 1 moderating the Mediator-DV path and direct IV-DV path

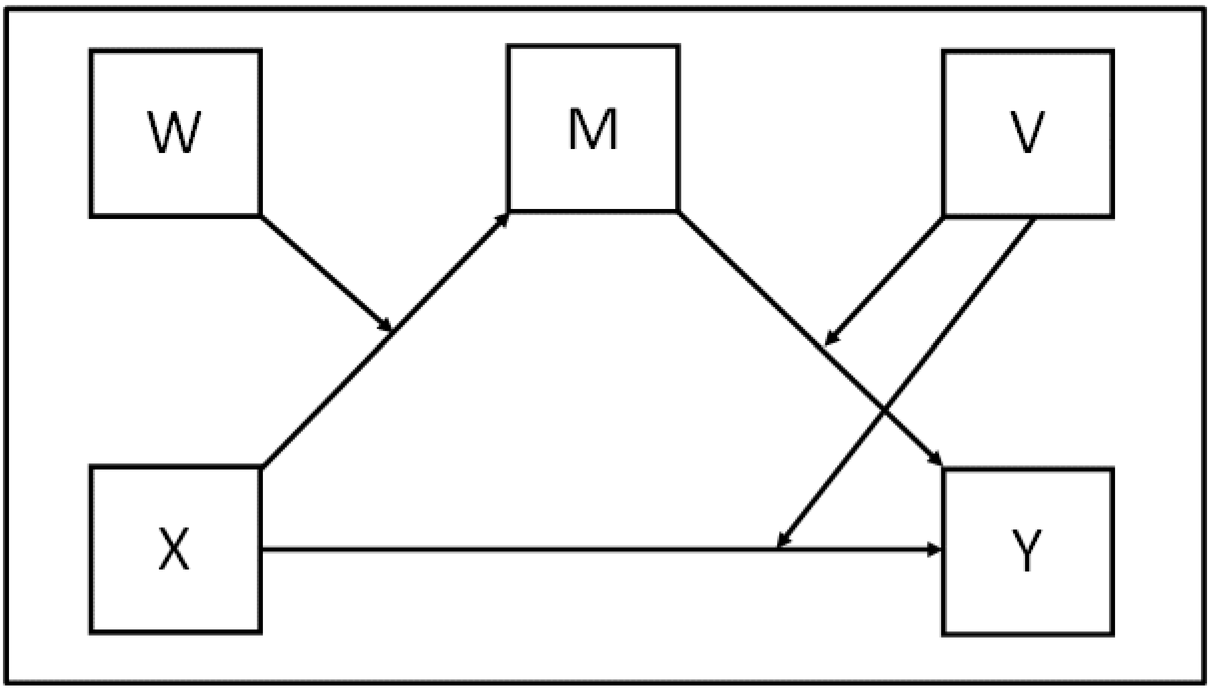
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

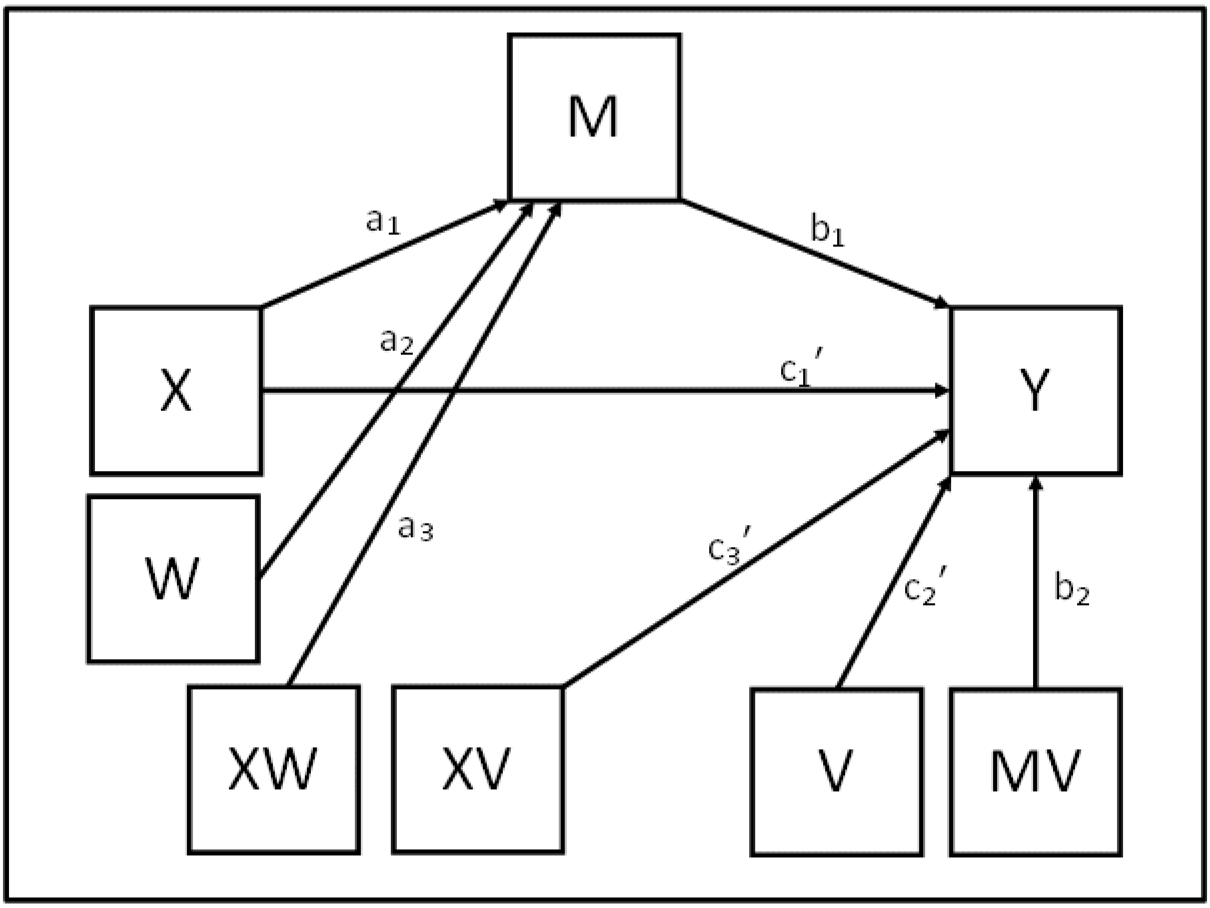
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2(a0 + a1X + a2W + a3XW)V + c1'X + c2'V + c3'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + a0b2V + a1b2XV + a2b2WV + a3b2XWV + c1'X + c2'V + c3'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a0b2V + a2b2WV + c2'V) + (a1b1 + a3b1W + a1b2V + a3b2WV + c1' + c3'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b2V + a3b2WV = (a1 + a3W)(b1 + b2V)

One direct effect of X on Y, conditional on V:

c1' + c3'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW XV MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
    MV = M*V;
    XW = X*W;
    XV = X*V;

ANALYSIS:
    TYPE = GENERAL;
    ESTIMATOR = ML;
    BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON MV (b2);

    Y ON X (cdash1);
    Y ON V (cdash2);
    Y ON XV (cdash3);
```

```
   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
    ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
    ILOW_HIV IMEW_HIV IHIW_HIV
    DIR_LOWV DIR_MEDV DIR_HIV
    TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
    TLOW_HIV TMEW_HIV THIW_HIV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V;
    IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V;
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V;

    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
```

```
a3*b2*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V;
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWV = cdash1 + cdash3*LOW_V;
    DIR_MEDV = cdash1 + cdash3*MED_V;
    DIR_HIV = cdash1 + cdash3*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + DIR_LOWV;
    TMEW_LOV = IMEW_LOV + DIR_LOWV;
    THIW_LOV = IHIW_LOV + DIR_LOWV;

    TLOW_MEV = ILOW_MEV + DIR_MEDV;
    TMEW_MEV = IMEW_MEV + DIR_MEDV;
    THIW_MEV = IHIW_MEV + DIR_MEDV;

    TLOW_HIV = ILOW_HIV + DIR_HIV;
    TMEW_HIV = IMEW_HIV + DIR_HIV;
    THIW_HIV = IHIW_HIV + DIR_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;
```

```
    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 29: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, 1 moderating the IV-Mediator path, 1 moderating the Mediator-DV path, both moderating the direct IV-DV path

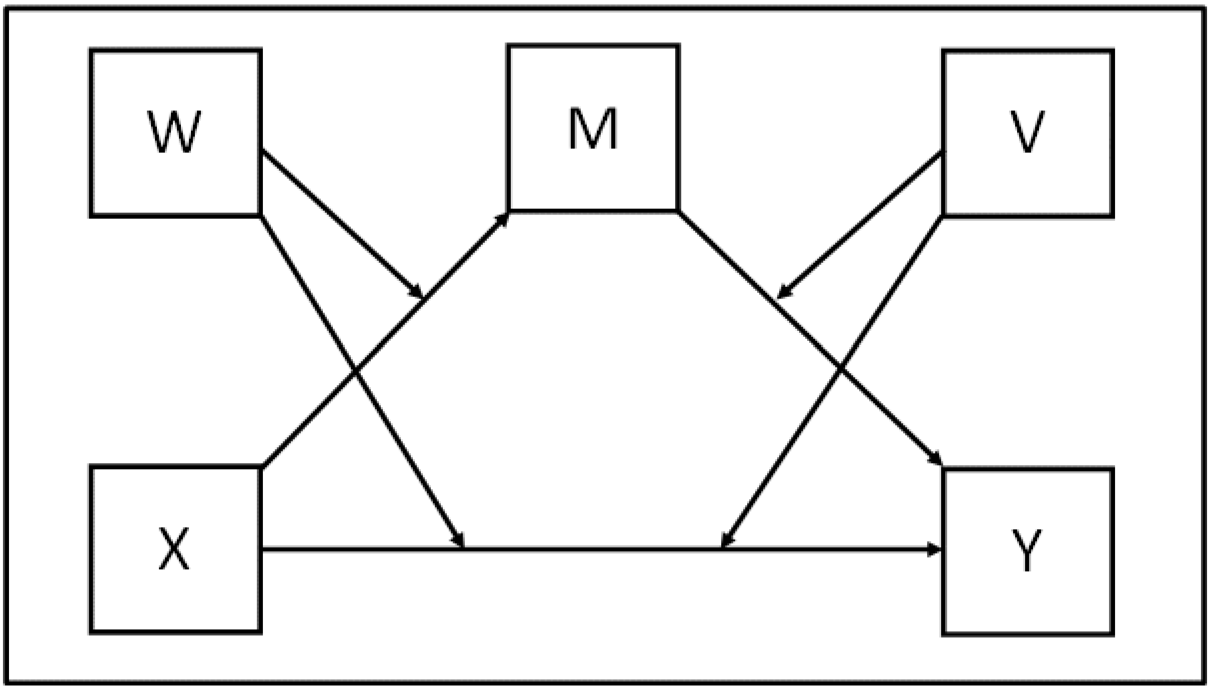
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

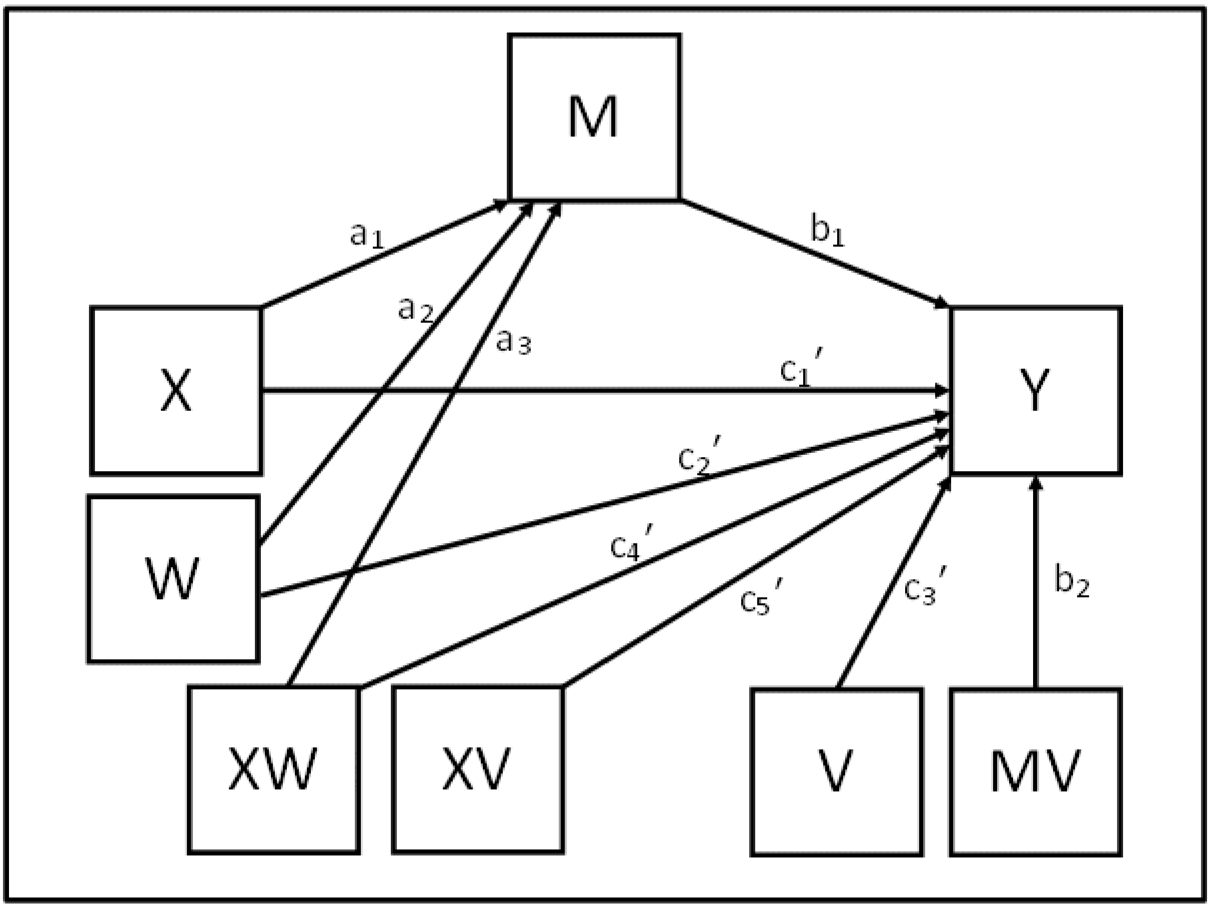
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + c1'X + c2'W + c3'V + c4'XW + c5'XV
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + c1'X + c2'W + c3'V + c4'XW + c5'XV
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2(a0 + a1X + a2W + a3XW)V + c1'X + c2'W + c3'V + c4'XW + c5'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + a0b2V + a1b2XV + a2b2WV + a3b2XWV + c1'X + c2'W + c3'V + c4'XW + c5'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a0b2V + a2b2WV + c2'W + c3'V) + (a1b1 + a3b1W + a1b2V + a3b2WV + c1' + c4'W + c5'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b2V + a3b2WV = (a1 + a3W)(b1 + b2V)

One direct effect of X on Y, conditional on W, V:

c1' + c4'W + c5'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW XV MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MV (b2);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON V (cdash3);
```

```
    Y ON XW (cdash4);
    Y ON XV (cdash5);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
    ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
    ILOW_HIV IMEW_HIV IHIW_HIV
    DLOW_LOV DMEW_LOV DHIW_LOV DLOW_MEV DMEW_MEV DHIW_MEV
    DLOW_HIV DMEW_HIV DHIW_HIV
    TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
    TLOW_HIV TMEW_HIV THIW_HIV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V;
    IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V;
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V;
```

```
    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V;
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOV = cdash1 + cdash4*LOW_W + cdash5*LOW_V;
    DMEW_LOV = cdash1 + cdash4*MED_W + cdash5*LOW_V;
    DHIW_LOV = cdash1 + cdash4*HIGH_W + cdash5*LOW_V;

    DLOW_MEV = cdash1 + cdash4*LOW_W + cdash5*MED_V;
    DMEW_MEV = cdash1 + cdash4*MED_W + cdash5*MED_V;
    DHIW_MEV = cdash1 + cdash4*HIGH_W + cdash5*MED_V;

    DLOW_HIV = cdash1 + cdash4*LOW_W + cdash5*HIGH_V;
    DMEW_HIV = cdash1 + cdash4*MED_W + cdash5*HIGH_V;
    DHIW_HIV = cdash1 + cdash4*HIGH_W + cdash5*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + DLOW_LOV;
    TMEW_LOV = IMEW_LOV + DMEW_LOV;
    THIW_LOV = IHIW_LOV + DHIW_LOV;

    TLOW_MEV = ILOW_MEV + DLOW_MEV;
    TMEW_MEV = IMEW_MEV + DMEW_MEV;
    THIW_MEV = IHIW_MEV + DHIW_MEV;

    TLOW_HIV = ILOW_HIV + DLOW_HIV;
    TMEW_HIV = IMEW_HIV + DMEW_HIV;
    THIW_HIV = IHIW_HIV + DHIW_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis
```

```
    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;

    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 30: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path, 1 moderating both the Mediator-DV path and the direct IV-DV path

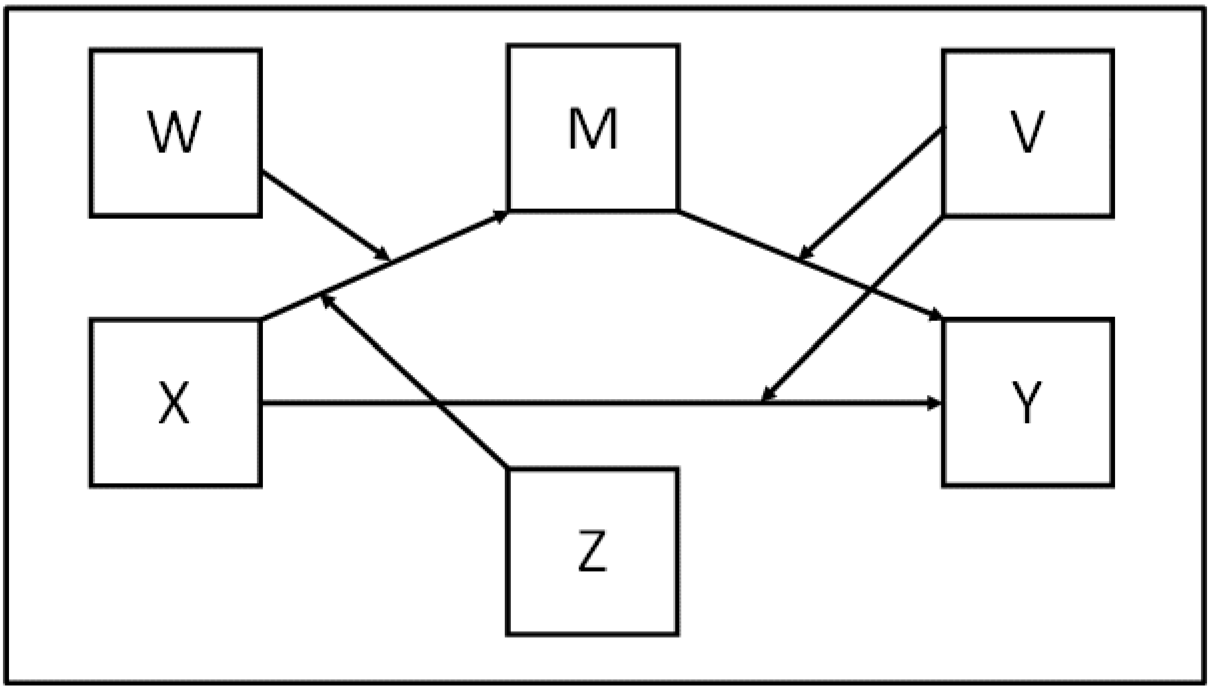
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

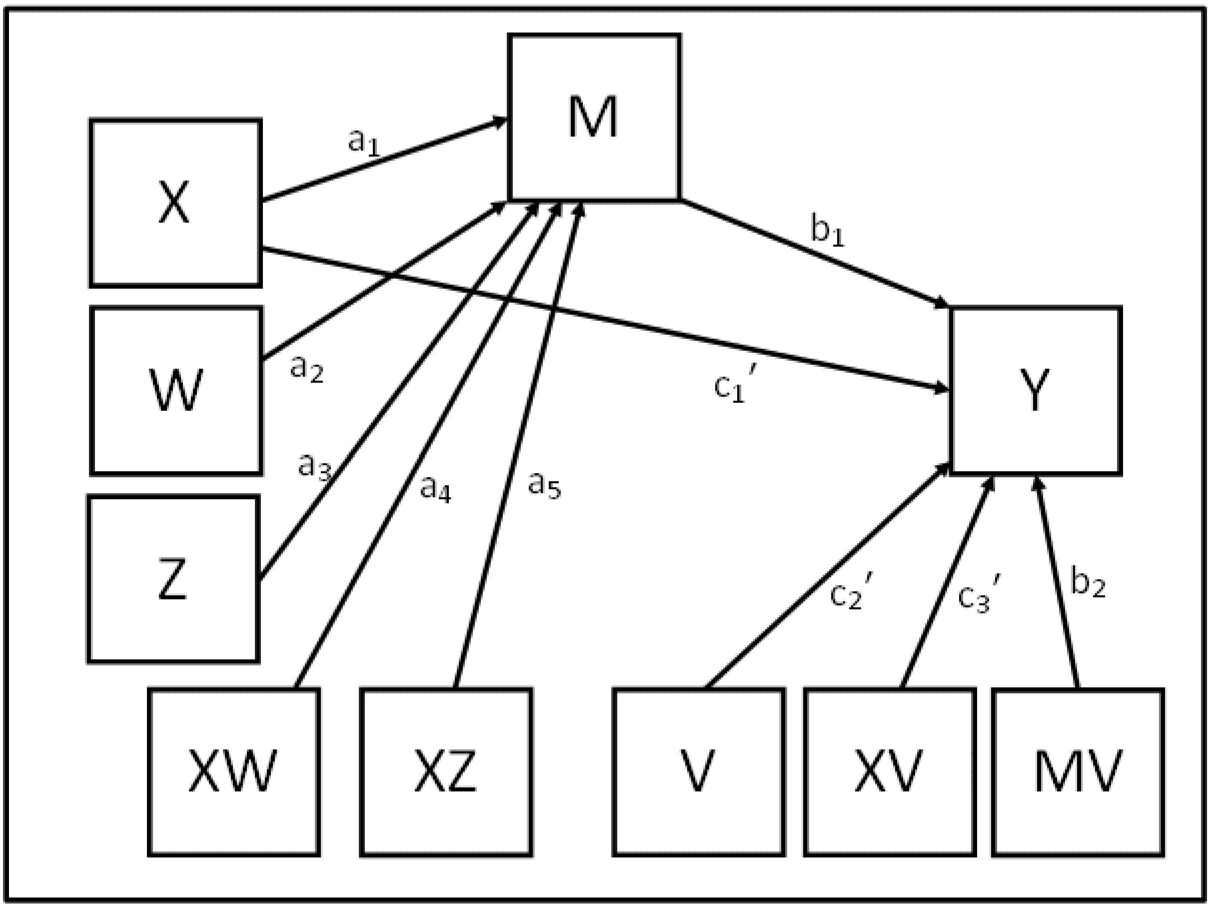
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ)V + c1'X + c2'V + c3'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a0b2V + a1b2XV + a2b1WV + a3b2ZV + a4b2XWV + a5b2XZV + c1'X + c2'V + c3'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a0b2V + a2b1WV + a3b2ZV + c2'V) + (a1b1 + a4b1W + a5b1Z + a1b2V + a4b2WV + a5b2ZV + c1' + c3'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V:

a1b1 + a4b1W + a5b1Z + a1b2V + a4b2WV + a5b2ZV = (a1 + a4W + a5Z) (b1 + b2V)

One direct effect of X on Y, conditional on V: c1' + c3'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ XV MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MV (b2);

   Y ON X (cdash1);
   Y ON V (cdash2);
   Y ON XV (cdash3);
```

```
   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
    ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
    ILWHZLV IMWHZLV IHWHZLV
    ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
    ILWHZMV IMWHZMV IHWHZMV
    ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
    ILWHZHV IMWHZHV IHWHZHV
    DIR_LOWV DIR_MEDV DIR_HIV
    TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
    TLWHZLV TMWHZLV THWHZLV
    TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
    TLWHZMV TMWHZMV THWHZMV
    TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
    TLWHZHV TMWHZHV THWHZHV);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z
```

```
    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
     a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
     a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_V
+
     a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*LOW_V
+
     a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V;
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b2*LOW_V +
     a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*MED_V
+
     a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_V
+
     a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V;
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V;
```

```
    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*MED_V
+
     a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_V
+
     a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V;

    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V;
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b2*MED_V +
     a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V;

    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a1*b2*HIGH_V +
     a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V;

    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b2*HIGH_V +
     a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V;

    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b2*HIGH_V +
     a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b2*HIGH_V +
     a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b2*HIGH_V +
     a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V;
```

```
! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWV = cdash1 + cdash3*LOW_V;
    DIR_MEDV = cdash1 + cdash3*MED_V;
    DIR_HIV = cdash1 + cdash3*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLWLZLV = ILWLZLV + DIR_LOWV;
    TMWLZLV = IMWLZLV + DIR_LOWV;
    THWLZLV = IHWLZLV + DIR_LOWV;

    TLWMZLV = ILWMZLV + DIR_LOWV;
    TMWMZLV = IMWMZLV + DIR_LOWV;
    THWMZLV = IHWMZLV + DIR_LOWV;

    TLWHZLV = ILWHZLV + DIR_LOWV;
    TMWHZLV = IMWHZLV + DIR_LOWV;
    THWHZLV = IHWHZLV + DIR_LOWV;

    TLWLZMV = ILWLZMV + DIR_MEDV;
    TMWLZMV = IMWLZMV + DIR_MEDV;
    THWLZMV = IHWLZMV + DIR_MEDV;

    TLWMZMV = ILWMZMV + DIR_MEDV;
    TMWMZMV = IMWMZMV + DIR_MEDV;
    THWMZMV = IHWMZMV + DIR_MEDV;

    TLWHZMV = ILWHZMV + DIR_MEDV;
    TMWHZMV = IMWHZMV + DIR_MEDV;
    THWHZMV = IHWHZMV + DIR_MEDV;

    TLWLZHV = ILWLZHV + DIR_HIV;
    TMWLZHV = IMWLZHV + DIR_HIV;
    THWLZHV = IHWLZHV + DIR_HIV;

    TLWMZHV = ILWMZHV + DIR_HIV;
    TMWMZHV = IMWMZHV + DIR_HIV;
    THWMZHV = IHWMZHV + DIR_HIV;

    TLWHZHV = ILWHZHV + DIR_HIV;
    TMWHZHV = IMWHZHV + DIR_HIV;
    THWHZHV = IHWHZHV + DIR_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis
```

```
    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);

    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;

    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;

    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;

    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 31: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating both the IV- Mediator path and the direct IV-DV path, 1 moderating both the Mediator-DV path and the direct IV-DV path

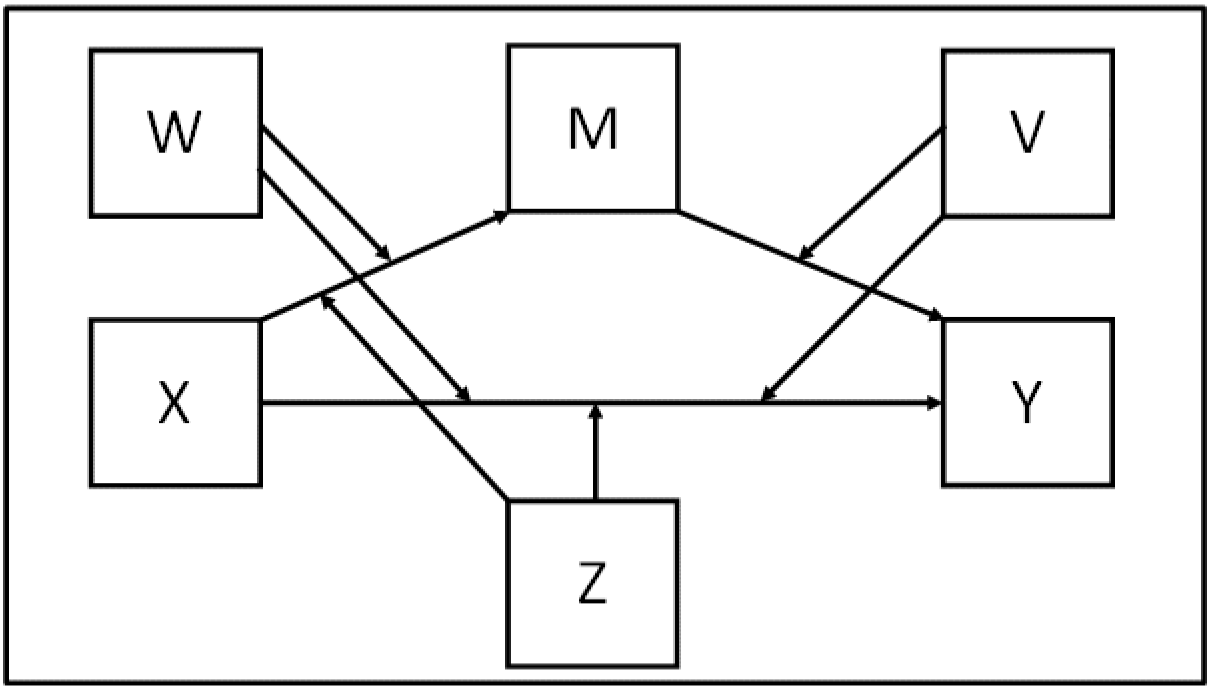
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

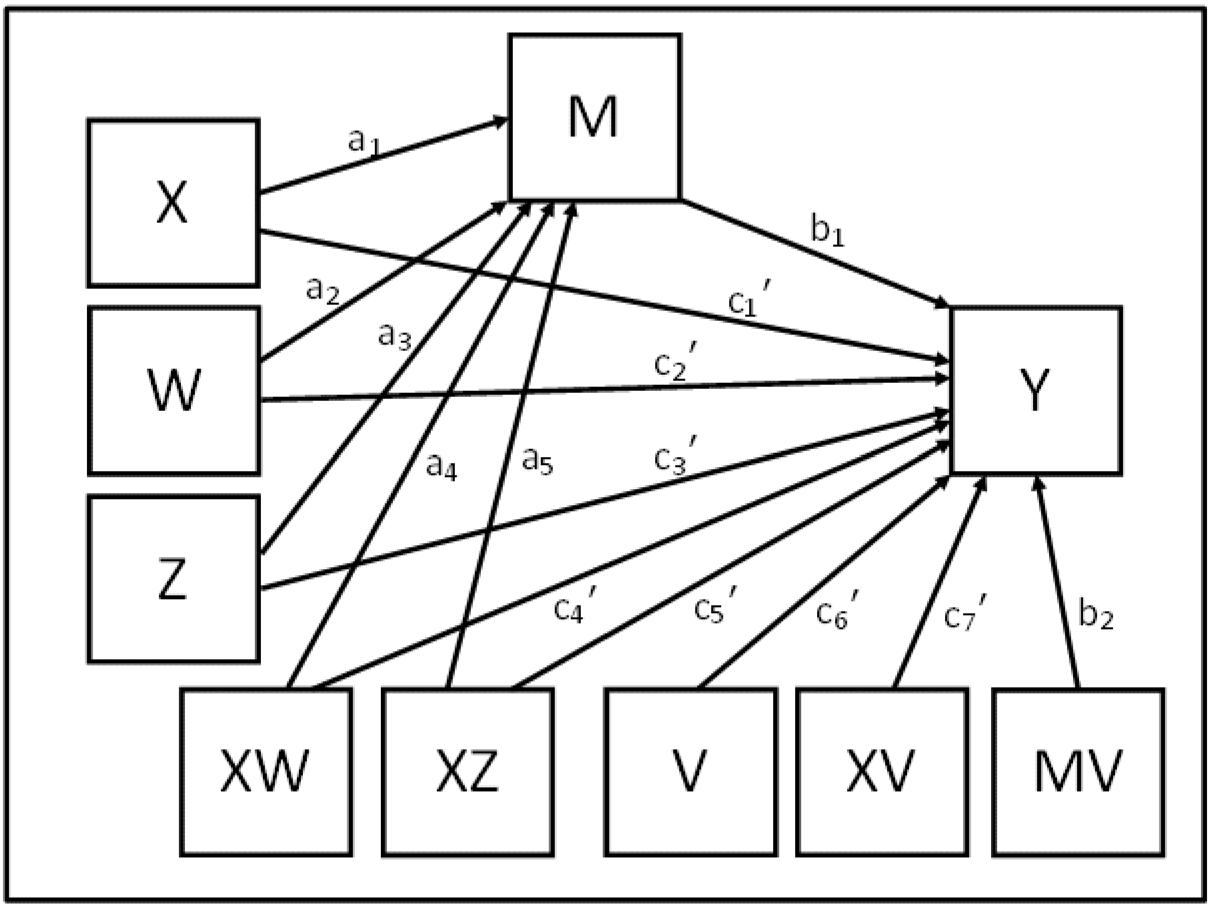
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'V + c7'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'V + c7'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ)V + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'V + c7'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a0b2V + a1b2XV + a2b2WV + a3b2ZV + a4b2XWV + a5b2XZV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'V + c7'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a0b2V + a2b2WV + a3b2ZV + c2'W + c3'Z + c6'V) + (a1b1 + a4b1W + a5b1Z + a1b2V + a4b2WV + a5b2ZV + c1' + c4'W + c5'Z + c7'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V:

a1b1 + a4b1W + a5b1Z + a1b2V + a4b2WV + a5b2ZV = (a1 + a4W + a5Z)(b1 + b2V)

One direct effect of X on Y, conditional on W, Z, V:

c1' + c4'W + c5'Z + c7'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ XV MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
```

```
    Y ON M (b1);
    Y ON MV (b2);

    Y ON X (cdash1);
    Y ON W (cdash2);
    Y ON Z (cdash3);
    Y ON XW (cdash4);
    Y ON XZ (cdash5);
    Y ON V (cdash6);
    Y ON XV (cdash7);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
    ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
    ILWHZLV IMWHZLV IHWHZLV
    ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
    ILWHZMV IMWHZMV IHWHZMV
    ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
    ILWHZHV IMWHZHV IHWHZHV
    DLWLZLV DMWLZLV DHWLZLV DLWMZLV DMWMZLV DHWMZLV
    DLWHZLV DMWHZLV DHWHZLV
    DLWLZMV DMWLZMV DHWLZMV DLWMZMV DMWMZMV DHWMZMV
    DLWHZMV DMWHZMV DHWHZMV
    DLWLZHV DMWLZHV DHWLZHV DLWMZHV DMWMZHV DHWMZHV
    DLWHZHV DMWHZHV DHWHZHV
    TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
    TLWHZLV TMWHZLV THWHZLV
    TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
    TLWHZMV TMWHZMV THWHZMV
```

```
    TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
    TLWHZHV TMWHZHV THWHZHV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
     a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
     a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_V
+
     a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*LOW_V
+
     a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V;
```

```
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b2*LOW_V +
     a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*MED_V
+
     a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_V
+
     a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V;
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V;

    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*MED_V
+
     a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_V
+
     a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V;

    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V;
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b2*MED_V +
     a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V;

    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a1*b2*HIGH_V +
     a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V;

    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
```

```
     a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b2*HIGH_V +
     a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V;

    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b2*HIGH_V +
     a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b2*HIGH_V +
     a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b2*HIGH_V +
     a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DLWLZLV = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_V;
    DMWLZLV = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*LOW_V;
    DHWLZLV = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*LOW_V;

    DLWMZLV = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_V;
    DMWMZLV = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*LOW_V;
    DHWMZLV = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*LOW_V;

    DLWHZLV = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_V;
    DMWHZLV = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*LOW_V;
    DHWHZLV = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*LOW_V;

    DLWLZMV = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*MED_V;
    DMWLZMV = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_V;
    DHWLZMV = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*MED_V;
```

```
    DLWMZMV = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*MED_V;
    DMWMZMV = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_V;
    DHWMZMV = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*MED_V;

    DLWHZMV = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*MED_V;
    DMWHZMV = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_V;
    DHWHZMV = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*MED_V;

    DLWLZHV = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*HIGH_V;
    DMWLZHV = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*HIGH_V;
    DHWLZHV = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_V;

    DLWMZHV = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*HIGH_V;
    DMWMZHV = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*HIGH_V;
    DHWMZHV = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_V;

    DLWHZHV = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*HIGH_V;
    DMWHZHV = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*HIGH_V;
    DHWHZHV = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLWLZLV = ILWLZLV + DLWLZLV;
    TMWLZLV = IMWLZLV + DMWLZLV;
    THWLZLV = IHWLZLV + DHWLZLV;

    TLWMZLV = ILWMZLV + DLWMZLV;
    TMWMZLV = IMWMZLV + DMWMZLV;
    THWMZLV = IHWMZLV + DHWMZLV;

    TLWHZLV = ILWHZLV + DLWHZLV;
    TMWHZLV = IMWHZLV + DMWHZLV;
    THWHZLV = IHWHZLV + DHWHZLV;
```

```
    TLWLZMV = ILWLZMV + DLWLZMV;
    TMWLZMV = IMWLZMV + DMWLZMV;
    THWLZMV = IHWLZMV + DHWLZMV;

    TLWMZMV = ILWMZMV + DLWMZMV;
    TMWMZMV = IMWMZMV + DMWMZMV;
    THWMZMV = IHWMZMV + DHWMZMV;

    TLWHZMV = ILWHZMV + DLWHZMV;
    TMWHZMV = IMWHZMV + DMWHZMV;
    THWHZMV = IHWHZMV + DHWHZMV;

    TLWLZHV = ILWLZHV + DLWLZHV;
    TMWLZHV = IMWLZHV + DMWLZHV;
    THWLZHV = IHWLZHV + DHWLZHV;

    TLWMZHV = ILWMZHV + DLWMZHV;
    TMWMZHV = IMWMZHV + DMWMZHV;
    THWMZHV = IHWMZHV + DHWMZHV;

    TLWHZHV = ILWHZHV + DLWHZHV;
    TMWHZHV = IMWHZHV + DMWHZHV;
    THWHZHV = IHWHZHV + DHWHZHV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);

    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;
```

```
    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;

    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;

    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 32: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path with all 2-way and 3-way interactions, 1 moderating both the Mediator-DV path and the direct IV-DV path

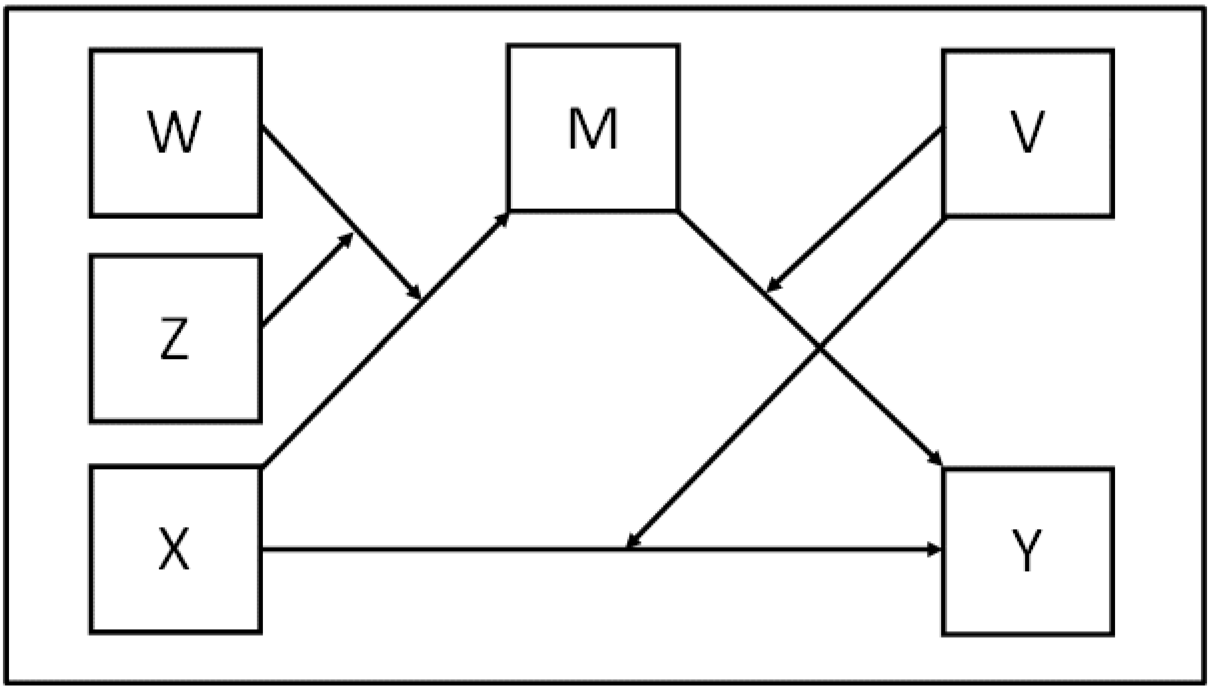
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

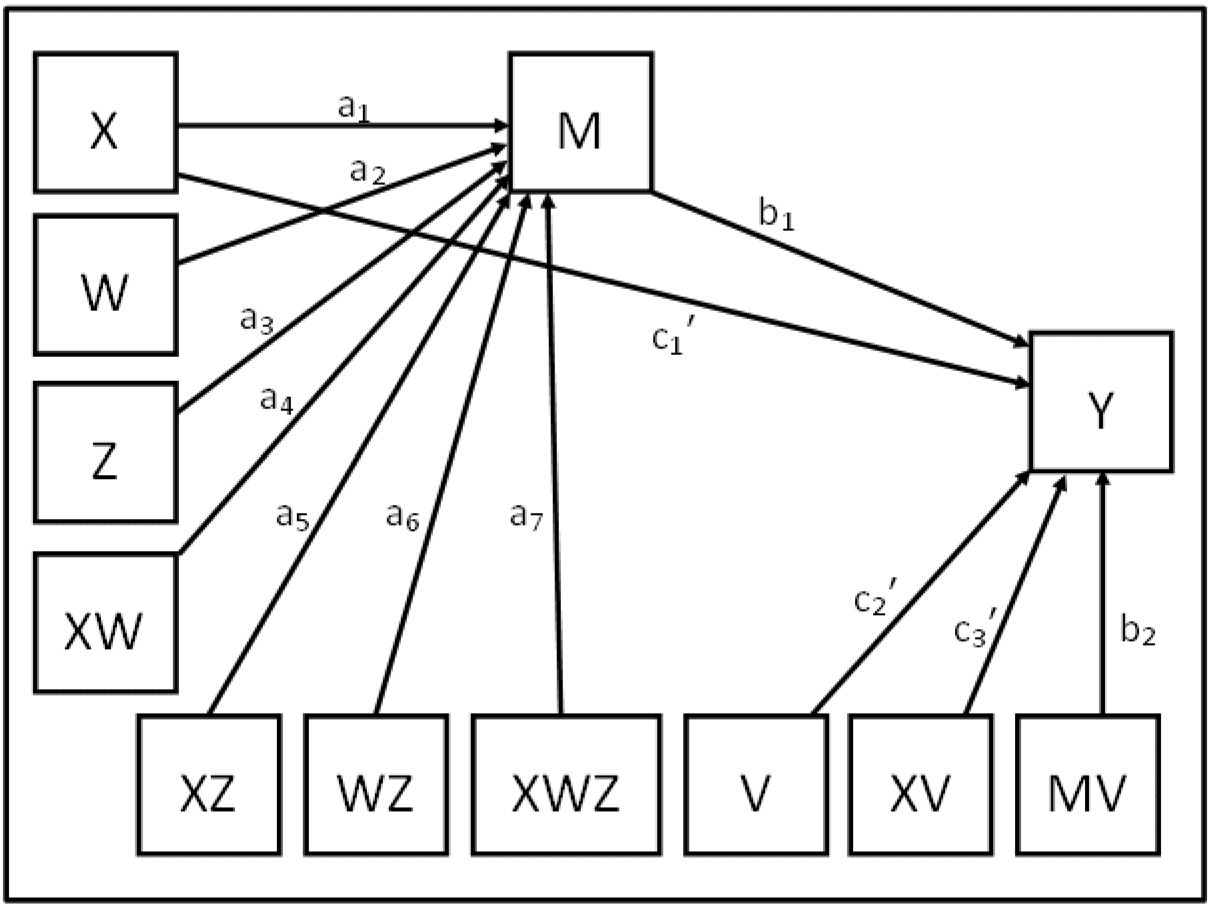
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + c1'X + c2'V + c3'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + a0b2 + a1b2XV + a2b2WV + a3b2ZV + a4b2XWV + a5b2XZV + a6b2WZV + a7b2XWZV + c1'X + c2'V + c3'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + a0b2 + a2b2WV + a3b2ZV + a6b2WZV + c2'V) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV + c1' + c3'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV = (a1 + a4W + a5Z + a7WZ)(b1 + b2V)

One direct effect of X on Y, conditional on V:

c1' + c3'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ WZ XV MV XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   XV = X*V;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MV (b2);

   Y ON X (cdash1);
   Y ON V (cdash2);
   Y ON XV (cdash3);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
    ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
    ILWHZLV IMWHZLV IHWHZLV
    ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
    ILWHZMV IMWHZMV IHWHZMV
    ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
    ILWHZHV IMWHZHV IHWHZHV
    DIR_LOWV DIR_MEDV DIR_HIV
    TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
    TLWHZLV TMWHZLV THWHZLV
    TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
    TLWHZMV TMWHZMV THWHZMV
    TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
    TLWHZHV TMWHZHV THWHZHV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
```

```
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
```

```
     a7*b2*LOW_W*HIGH_Z*LOW_V;
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V;
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V;

    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V;

    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V;
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V;
```

```
    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V;

    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V;

    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values
```

```
    DIR_LOWV = cdash1 + cdash3*LOW_V;
    DIR_MEDV = cdash1 + cdash3*MED_V;
    DIR_HIV = cdash1 + cdash3*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLWLZLV = ILWLZLV + DIR_LOWV;
    TMWLZLV = IMWLZLV + DIR_LOWV;
    THWLZLV = IHWLZLV + DIR_LOWV;

    TLWMZLV = ILWMZLV + DIR_LOWV;
    TMWMZLV = IMWMZLV + DIR_LOWV;
    THWMZLV = IHWMZLV + DIR_LOWV;

    TLWHZLV = ILWHZLV + DIR_LOWV;
    TMWHZLV = IMWHZLV + DIR_LOWV;
    THWHZLV = IHWHZLV + DIR_LOWV;

    TLWLZMV = ILWLZMV + DIR_MEDV;
    TMWLZMV = IMWLZMV + DIR_MEDV;
    THWLZMV = IHWLZMV + DIR_MEDV;

    TLWMZMV = ILWMZMV + DIR_MEDV;
    TMWMZMV = IMWMZMV + DIR_MEDV;
    THWMZMV = IHWMZMV + DIR_MEDV;

    TLWHZMV = ILWHZMV + DIR_MEDV;
    TMWHZMV = IMWHZMV + DIR_MEDV;
    THWHZMV = IHWHZMV + DIR_MEDV;

    TLWLZHV = ILWLZHV + DIR_HIV;
    TMWLZHV = IMWLZHV + DIR_HIV;
    THWLZHV = IHWLZHV + DIR_HIV;

    TLWMZHV = ILWMZHV + DIR_HIV;
    TMWMZHV = IMWMZHV + DIR_HIV;
    THWMZHV = IHWMZHV + DIR_HIV;

    TLWHZHV = ILWHZHV + DIR_HIV;
    TMWHZHV = IMWHZHV + DIR_HIV;
    THWHZHV = IHWHZHV + DIR_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
```

```
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);

    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;

    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;

    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;

    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 33: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating both the IV- Mediator path and the direct IV-DV path with all 2-way and 3-way interactions, 1 moderating both the Mediator-DV path and the direct IV-DV path

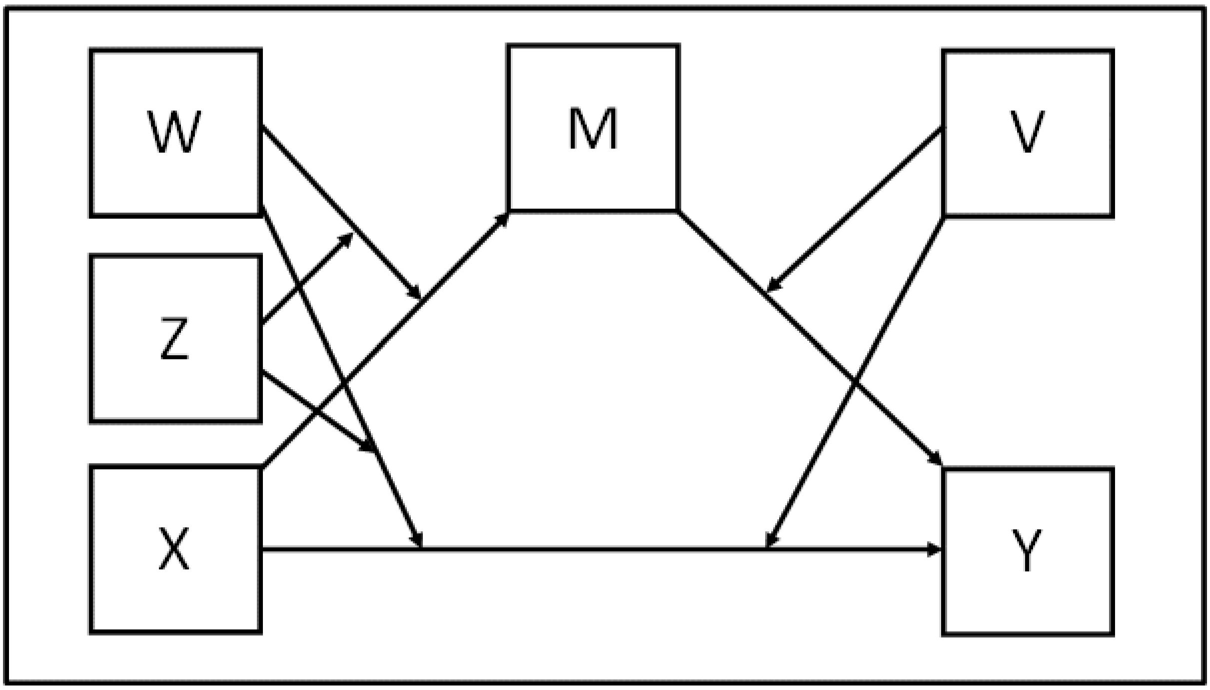
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

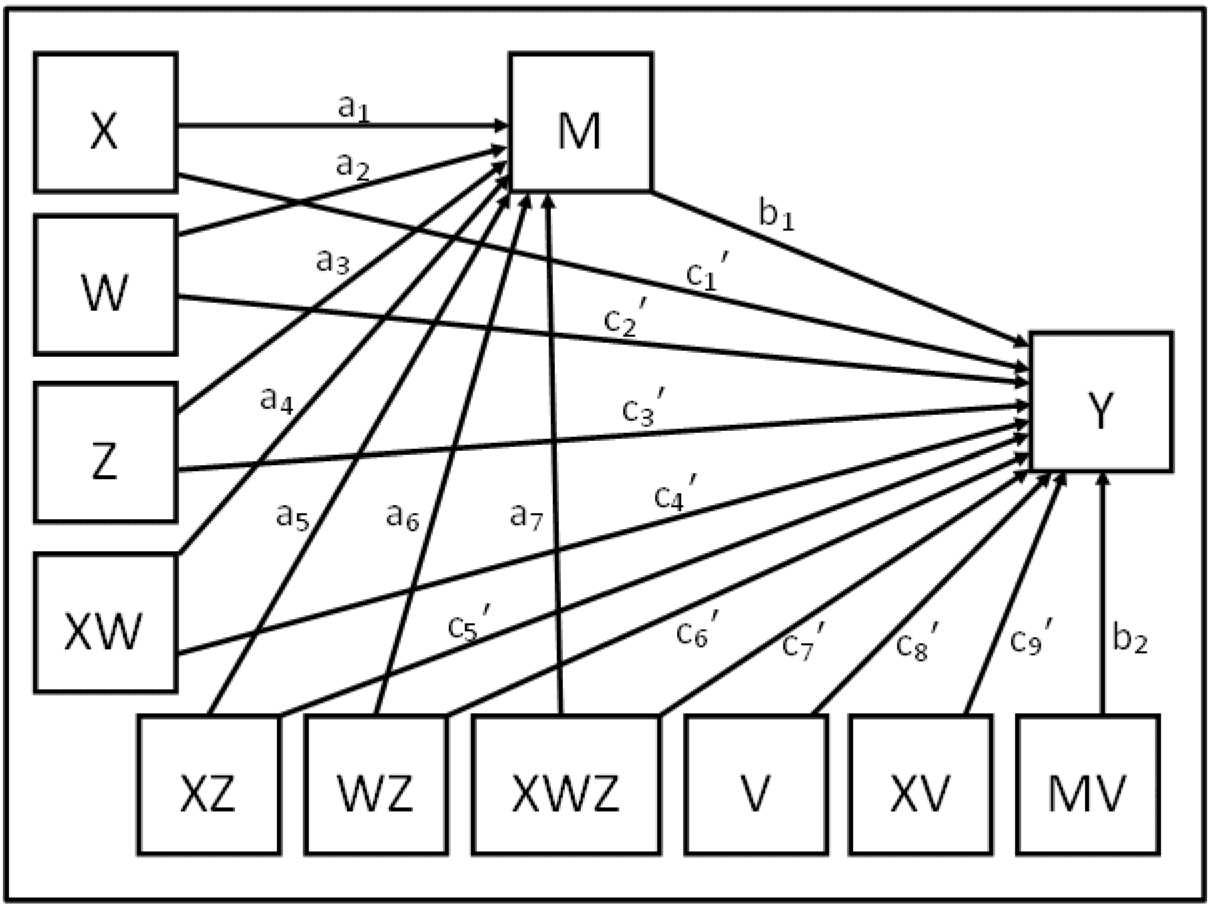
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ + c8'V + c9'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ + c8'V + c9'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ + c8'V + c9'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + a0b2V + a1b2XV + a2b2WV + a3b2ZV + a4b2XWV + a5b2XZV + a6b2WZV + a7b2XWZV + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ + c8'V + c9'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + a0b2V + a2b2WV + a3b2ZV + a6b2WZV + c2'W + c3'Z + c6'WZ + c8'V) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV + c1' + c4'W + c5'Z + c7'WZ + c9'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV = (a1 + a4W + a5Z + a7WZ)(b1 + b2V)

One direct effect of X on Y, conditional on W, Z, V:

c1' + c4'W + c5'Z + c7'WZ + c9'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ WZ XV MV XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   XV = X*V;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;
```

```
! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MV (b2);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON Z (cdash3);
   Y ON XW (cdash4);
   Y ON XZ (cdash5);
   Y ON WZ (cdash6);
   Y ON XWZ (cdash7);
   Y ON V (cdash8);
   Y ON XV (cdash9);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
    ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
    ILWHZLV IMWHZLV IHWHZLV
    ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
    ILWHZMV IMWHZMV IHWHZMV
    ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
    ILWHZHV IMWHZHV IHWHZHV
    DLWLZLV DMWLZLV DHWLZLV DLWMZLV DMWMZLV DHWMZLV
    DLWHZLV DMWHZLV DHWHZLV
```

```
    DLWLZMV DMWLZMV DHWLZMV DLWMZMV DMWMZMV DHWMZMV
    DLWHZMV DMWHZMV DHWHZMV
    DLWLZHV DMWLZHV DHWLZHV DLWMZHV DMWMZHV DHWMZHV
    DLWHZHV DMWHZHV DHWHZHV
    TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
    TLWHZLV TMWHZLV THWHZLV
    TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
    TLWHZMV TMWHZMV THWHZMV
    TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
    TLWHZHV TMWHZHV THWHZHV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
```

```
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V;
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V;
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V;

    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V;
```

```
    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V;
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V;

    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V;

    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V;

    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
```

```
+
     a7*b2*MED_W*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DLWLZLV = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash9*LOW_V;
    DMWLZLV = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash9*LOW_V;
    DHWLZLV = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash9*LOW_V;

    DLWMZLV = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash9*LOW_V;
    DMWMZLV = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash9*LOW_V;
    DHWMZLV = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash9*LOW_V;

    DLWHZLV = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash9*LOW_V;
    DMWHZLV = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash9*LOW_V;
    DHWHZLV = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash9*LOW_V;

    DLWLZMV = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash9*MED_V;
    DMWLZMV = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash9*MED_V;
    DHWLZMV = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
```

```
cdash7*HIGH_W*LOW_Z +
     cdash9*MED_V;

    DLWMZMV = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash9*MED_V;
    DMWMZMV = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash9*MED_V;
    DHWMZMV = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash9*MED_V;

    DLWHZMV = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash9*MED_V;
    DMWHZMV = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash9*MED_V;
    DHWHZMV = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash9*MED_V;

    DLWLZHV = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash9*HIGH_V;
    DMWLZHV = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash9*HIGH_V;
    DHWLZHV = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash9*HIGH_V;

    DLWMZHV = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash9*HIGH_V;
    DMWMZHV = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash9*HIGH_V;
    DHWMZHV = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash9*HIGH_V;

    DLWHZHV = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash9*HIGH_V;
    DMWHZHV = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash9*HIGH_V;
```

```
    DHWHZHV = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash9*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLWLZLV = ILWLZLV + DLWLZLV;
    TMWLZLV = IMWLZLV + DMWLZLV;
    THWLZLV = IHWLZLV + DHWLZLV;

    TLWMZLV = ILWMZLV + DLWMZLV;
    TMWMZLV = IMWMZLV + DMWMZLV;
    THWMZLV = IHWMZLV + DHWMZLV;

    TLWHZLV = ILWHZLV + DLWHZLV;
    TMWHZLV = IMWHZLV + DMWHZLV;
    THWHZLV = IHWHZLV + DHWHZLV;

    TLWLZMV = ILWLZMV + DLWLZMV;
    TMWLZMV = IMWLZMV + DMWLZMV;
    THWLZMV = IHWLZMV + DHWLZMV;

    TLWMZMV = ILWMZMV + DLWMZMV;
    TMWMZMV = IMWMZMV + DMWMZMV;
    THWMZMV = IHWMZMV + DHWMZMV;

    TLWHZMV = ILWHZMV + DLWHZMV;
    TMWHZMV = IMWHZMV + DMWHZMV;
    THWHZMV = IHWHZMV + DHWHZMV;

    TLWLZHV = ILWLZHV + DLWLZHV;
    TMWLZHV = IMWLZHV + DMWLZHV;
    THWLZHV = IHWLZHV + DHWLZHV;

    TLWMZHV = ILWMZHV + DLWMZHV;
    TMWMZHV = IMWMZHV + DMWMZHV;
    THWMZHV = IHWMZHV + DHWMZHV;

    TLWHZHV = ILWHZHV + DLWHZHV;
    TMWHZHV = IMWHZHV + DMWHZHV;
    THWHZHV = IHWHZHV + DHWHZHV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
```

```
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);

    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;

    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;

    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;

    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 34: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 2 moderating the IV-Mediator path with all 2-way and 3-way interactions, one of which also moderates the direct IV-DV path, 1 moderating the Mediator-DV path and the direct IV-DV path

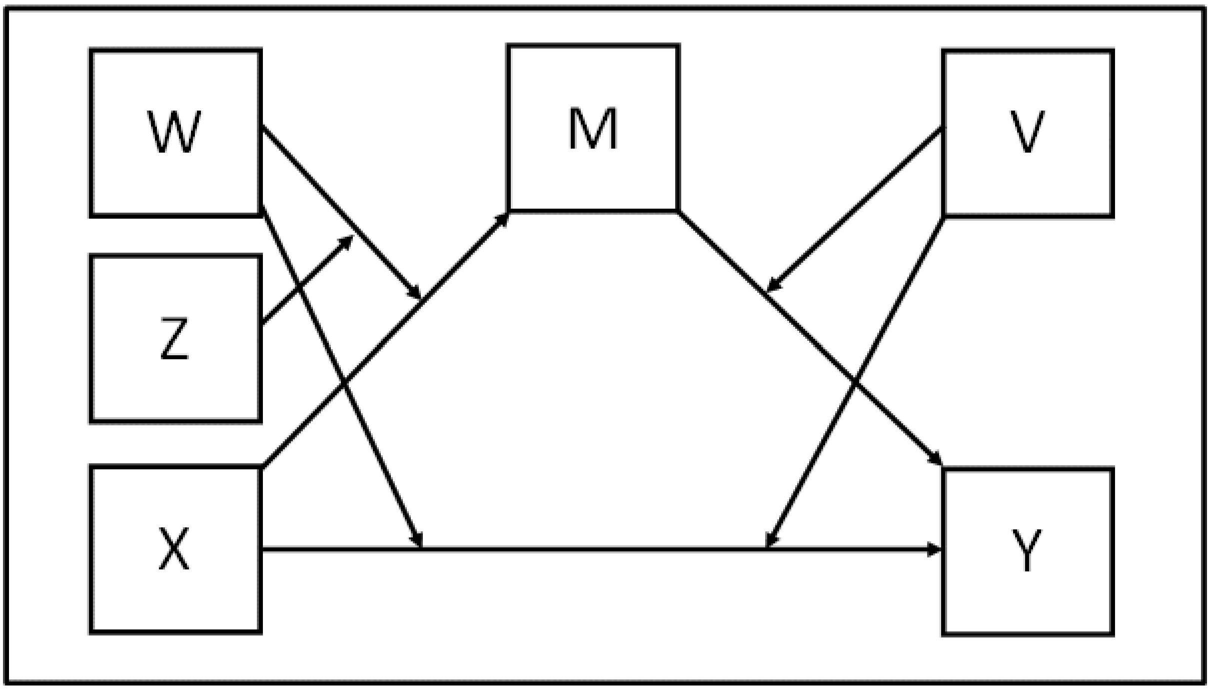
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, Z, V, 1 outcome Y

Preliminary notes:

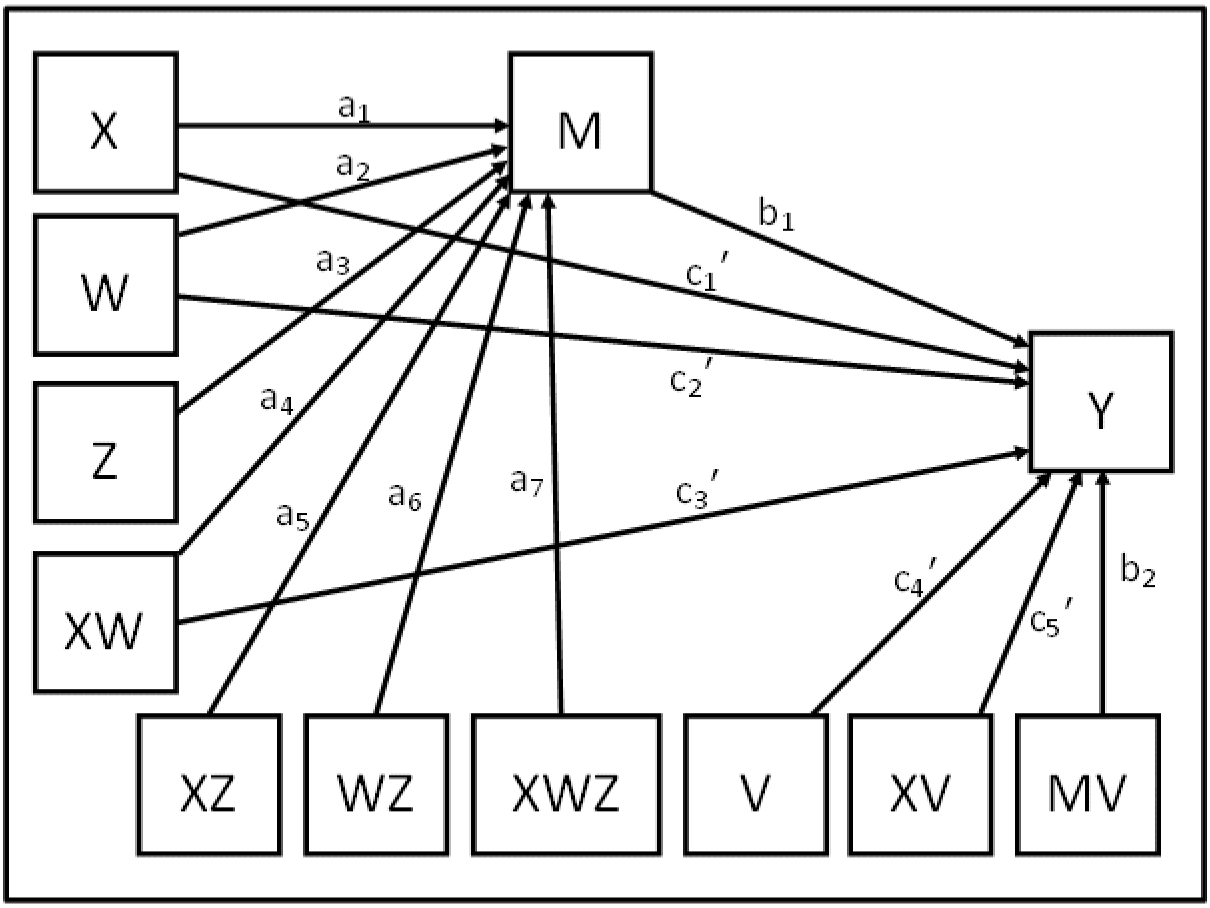
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + c1'X + c2'W + c3'XW + c4'V + c5'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + c1'X + c2'W + c3'XW + c4'V + c5'XV
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + c1'X + c2'W + c3'XW + c4'V + c5'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + a0b2V + a1b2XV + a2b2WV + a3b2ZV + a4b2XWV + a5b2XZV + a6b2WZV + a7b2XWZV + c1'X + c2'W + c3'XW + c4'V + c5'XV

Hence... grouping terms into form Y = a + bX
Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + a0b2V + a2b2WV + a3b2ZV + a6b2WZV + c2'W + c4'V) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV + c1' + c3'W + c5'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V: a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV = (a1 + a4W + a5Z + a7WZ)(b1 + b2V)

One direct effect of X on Y, conditional on W, V: c1' + c3'W + c5'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V
! Outcome variable - Y

USEVARIABLES = X M W Z V Y XW XZ WZ XV MV XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   XV = X*V;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MV (b2);
```

```
    Y ON X (cdash1);
    Y ON W (cdash2);
    Y ON XW (cdash3);
    Y ON V (cdash4);
    Y ON XV (cdash5);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
    M ON WZ (a6);
    M ON XWZ (a7);
! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V
! for example, of 1 SD below mean, mean, 1 SD above mean
! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.
MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V
    ILWLZLV IMWLZLV IHWLZLV ILWMZLV IMWMZLV IHWMZLV
    ILWHZLV IMWHZLV IHWHZLV
    ILWLZMV IMWLZMV IHWLZMV ILWMZMV IMWMZMV IHWMZMV
    ILWHZMV IMWHZMV IHWHZMV
    ILWLZHV IMWLZHV IHWLZHV ILWMZHV IMWMZHV IHWMZHV
    ILWHZHV IMWHZHV IHWHZHV
    DLOW_LOV DMEW_LOV DHIW_LOV DLOW_MEV DMEW_MEV DHIW_MEV
    DLOW_HIV DMEW_HIV DHIW_HIV
    TLWLZLV TMWLZLV THWLZLV TLWMZLV TMWMZLV THWMZLV
    TLWHZLV TMWHZLV THWHZLV
    TLWLZMV TMWLZMV THWLZMV TLWMZMV TMWMZMV THWMZMV
    TLWHZMV TMWHZMV THWHZMV
    TLWLZHV TMWLZHV THWLZHV TLWMZHV TMWMZHV THWMZHV
    TLWHZHV TMWHZHV THWHZHV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W
```

```
    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILWLZLV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V;
    IMWLZLV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V;
    IHWLZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V;

    ILWMZLV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V;
    IMWMZLV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V;
    IHWMZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V;

    ILWHZLV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V;
    IMWHZLV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
```

```
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V;
    IHWHZLV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V;

    ILWLZMV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V;
    IMWLZMV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V;
    IHWLZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V;

    ILWMZMV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V;
    IMWMZMV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V;
    IHWMZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V;

    ILWHZMV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V;
    IMWHZMV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V;
    IHWHZMV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V;

    ILWLZHV = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
```

```
     a7*b2*LOW_W*LOW_Z*HIGH_V;
    IMWLZHV = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V;
    IHWLZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V;

    ILWMZHV = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V;
    IMWMZHV = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V;
    IHWMZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V;

    ILWHZHV = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V;
    IMWHZHV = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V;
    IHWHZHV = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOV = cdash1 + cdash3*LOW_W + cdash5*LOW_V;
    DMEW_LOV = cdash1 + cdash3*MED_W + cdash5*LOW_V;
    DHIW_LOV = cdash1 + cdash3*HIGH_W + cdash5*LOW_V;
```

```
    DLOW_MEV = cdash1 + cdash3*LOW_W + cdash5*MED_V;
    DMEW_MEV = cdash1 + cdash3*MED_W + cdash5*MED_V;
    DHIW_MEV = cdash1 + cdash3*HIGH_W + cdash5*MED_V;

    DLOW_HIV = cdash1 + cdash3*LOW_W + cdash5*HIGH_V;
    DMEW_HIV = cdash1 + cdash3*MED_W + cdash5*HIGH_V;
    DHIW_HIV = cdash1 + cdash3*HIGH_W + cdash5*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLWLZLV = ILWLZLV + DLOW_LOV;
    TMWLZLV = IMWLZLV + DMEW_LOV;
    THWLZLV = IHWLZLV + DHIW_LOV;

    TLWMZLV = ILWMZLV + DLOW_LOV;
    TMWMZLV = IMWMZLV + DMEW_LOV;
    THWMZLV = IHWMZLV + DHIW_LOV;

    TLWHZLV = ILWHZLV + DLOW_LOV;
    TMWHZLV = IMWHZLV + DMEW_LOV;
    THWHZLV = IHWHZLV + DHIW_LOV;

    TLWLZMV = ILWLZMV + DLOW_MEV;
    TMWLZMV = IMWLZMV + DMEW_MEV;
    THWLZMV = IHWLZMV + DHIW_MEV;

    TLWMZMV = ILWMZMV + DLOW_MEV;
    TMWMZMV = IMWMZMV + DMEW_MEV;
    THWMZMV = IHWMZMV + DHIW_MEV;

    TLWHZMV = ILWHZMV + DLOW_MEV;
    TMWHZMV = IMWHZMV + DMEW_MEV;
    THWHZMV = IHWHZMV + DHIW_MEV;

    TLWLZHV = ILWLZHV + DLOW_HIV;
    TMWLZHV = IMWLZHV + DMEW_HIV;
    THWLZHV = IHWLZHV + DHIW_HIV;

    TLWMZHV = ILWMZHV + DLOW_HIV;
    TMWMZHV = IMWMZHV + DMEW_HIV;
    THWMZHV = IHWMZHV + DHIW_HIV;

    TLWHZHV = ILWHZHV + DLOW_HIV;
    TMWHZHV = IMWHZHV + DMEW_HIV;
    THWHZHV = IHWHZHV + DHIW_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be
```

```
replaced ! by logical min and max limits of predictor X used
in analysis
    PLOT(PLWLZLV PMWLZLV PHWLZLV PLWMZLV PMWMZLV PHWMZLV
    PLWHZLV PMWHZLV PHWHZLV
    PLWLZMV PMWLZMV PHWLZMV PLWMZMV PMWMZMV PHWMZMV
    PLWHZMV PMWHZMV PHWHZMV
    PLWLZHV PMWLZHV PHWLZHV PLWMZHV PMWMZHV PHWMZHV
    PLWHZHV PMWHZHV PHWHZHV);

    LOOP(XVAL,1,5,0.1);

    PLWLZLV = ILWLZLV*XVAL;
    PMWLZLV = IMWLZLV*XVAL;
    PHWLZLV = IHWLZLV*XVAL;

    PLWMZLV = ILWMZLV*XVAL;
    PMWMZLV = IMWMZLV*XVAL;
    PHWMZLV = IHWMZLV*XVAL;

    PLWHZLV = ILWHZLV*XVAL;
    PMWHZLV = IMWHZLV*XVAL;
    PHWHZLV = IHWHZLV*XVAL;

    PLWLZMV = ILWLZMV*XVAL;
    PMWLZMV = IMWLZMV*XVAL;
    PHWLZMV = IHWLZMV*XVAL;

    PLWMZMV = ILWMZMV*XVAL;
    PMWMZMV = IMWMZMV*XVAL;
    PHWMZMV = IHWMZMV*XVAL;

    PLWHZMV = ILWHZMV*XVAL;
    PMWHZMV = IMWHZMV*XVAL;
    PHWHZMV = IHWHZMV*XVAL;

    PLWLZHV = ILWLZHV*XVAL;
    PMWLZHV = IMWLZHV*XVAL;
    PHWLZHV = IHWLZHV*XVAL;

    PLWMZHV = ILWMZHV*XVAL;
    PMWMZHV = IMWMZHV*XVAL;
    PHWMZHV = IHWMZHV*XVAL;

    PLWHZHV = ILWHZHV*XVAL;
    PMWHZHV = IMWHZHV*XVAL;
    PHWHZHV = IHWHZHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 35: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating the Mediator-DV path

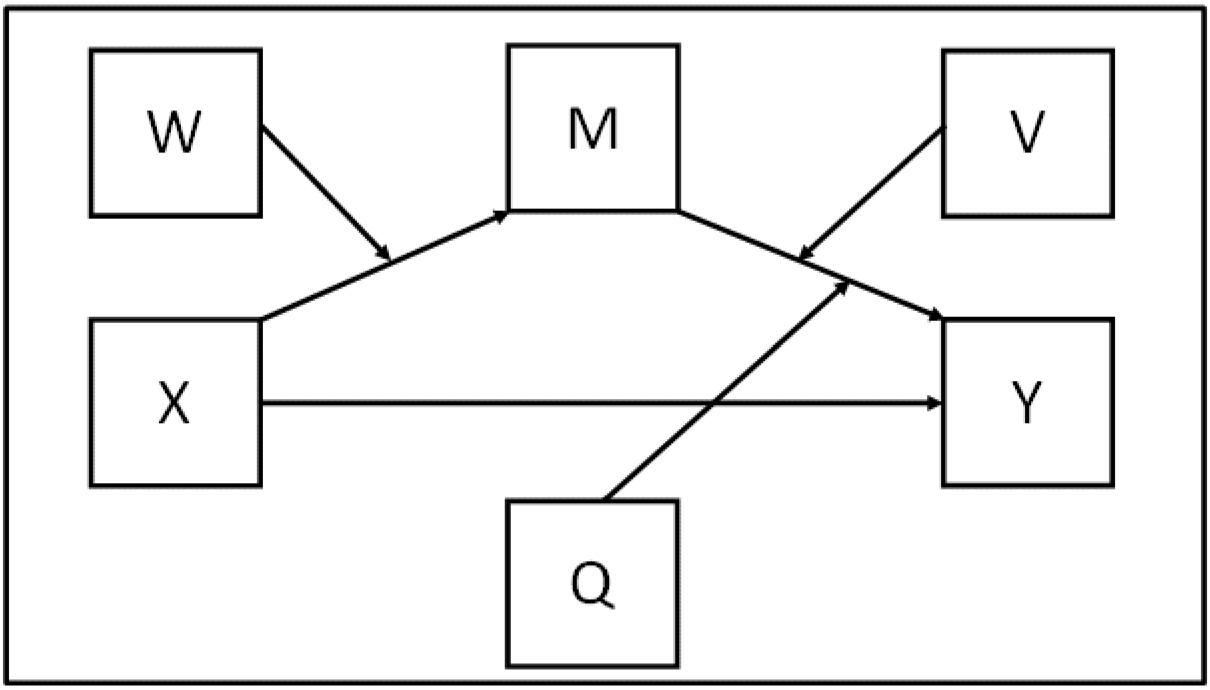
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

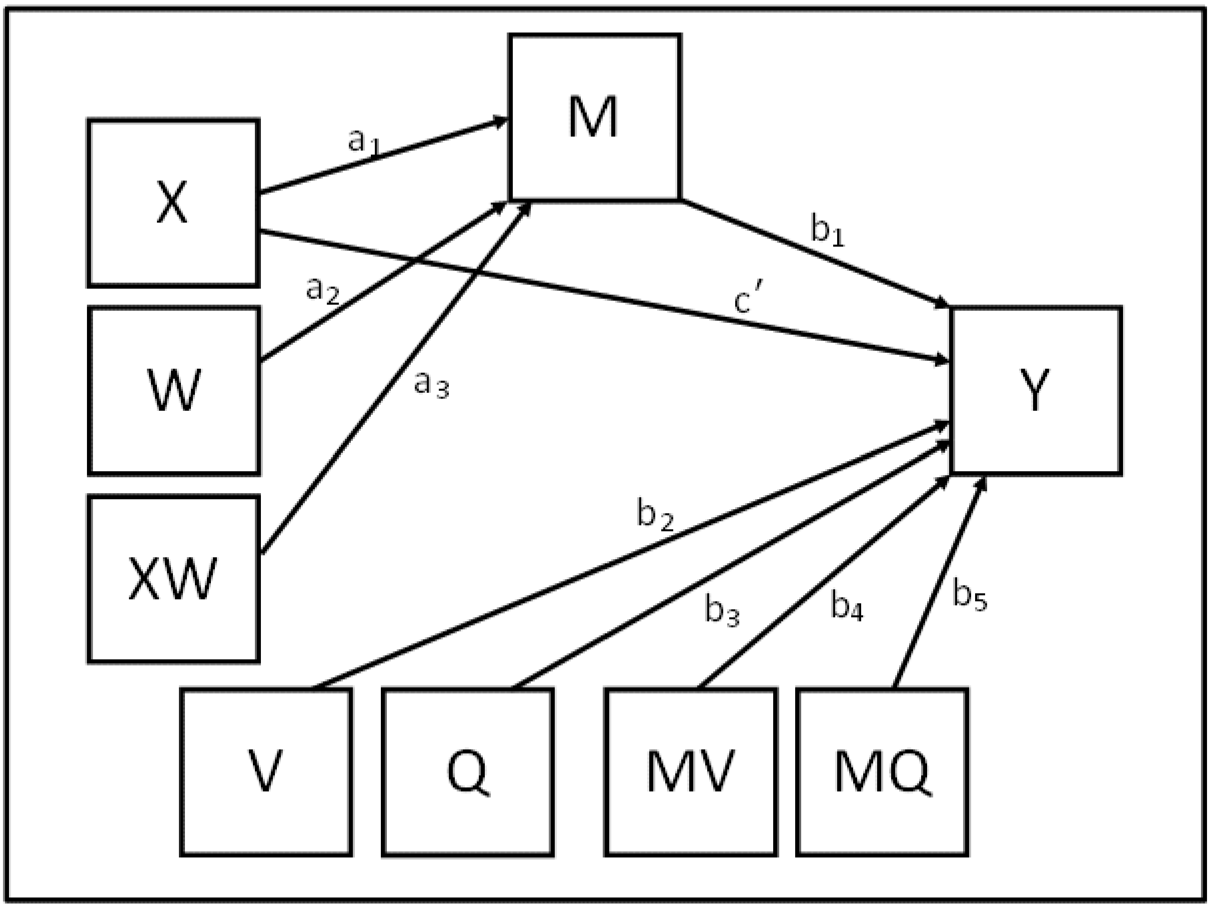
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c'X
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c'X
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2V + b3Q + b4(a0 + a1X + a2W + a3XW)V + b5(a0 + a1X + a2W + a3XW)Q + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4XWV + a0b5Q + a1b5XQ + a2b5WQ + a3b5XWQ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2V + b3Q + a0b4V + a2b4WV + a0b5Q + a2b5WQ) + (a1b1 + a3b1W + a1b4V + a3b4WV + a1b5Q + a3b5WQ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, V, Q:

a1b1 + a3b1W + a1b4V + a3b4WV + a1b5Q + a3b5WQ = (a1 + a3W)(b1 + b4V + b5Q)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) - W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW MV MQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);
```

```
    Y ON X (cdash);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
    ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
    ILWHVLQ IMWHVLQ IHWHVLQ
    ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
    ILWHVMQ IMWHVMQ IHWHVMQ
    ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
    ILWHVHQ IMWHVHQ IHWHVHQ
    TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
    TLWHVLQ TMWHVLQ THWHVLQ
    TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
    TLWHVMQ TMWHVMQ THWHVMQ
    TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
    TLWHVHQ TMWHVHQ THWHVHQ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V
```

```
    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q;

    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q;

    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q;
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q;

    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q;
```

```
    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q;
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q;

    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q;

    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q;

    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q;
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q;
```

```
! Calc conditional total effects for each combination of
moderator values

    TLWLVLQ = ILWLVLQ + cdash;
    TMWLVLQ = IMWLVLQ + cdash;
    THWLVLQ = IHWLVLQ + cdash;

    TLWMVLQ = ILWMVLQ + cdash;
    TMWMVLQ = IMWMVLQ + cdash;
    THWMVLQ = IHWMVLQ + cdash;

    TLWHVLQ = ILWHVLQ + cdash;
    TMWHVLQ = IMWHVLQ + cdash;
    THWHVLQ = IHWHVLQ + cdash;

    TLWLVMQ = ILWLVMQ + cdash;
    TMWLVMQ = IMWLVMQ + cdash;
    THWLVMQ = IHWLVMQ + cdash;

    TLWMVMQ = ILWMVMQ + cdash;
    TMWMVMQ = IMWMVMQ + cdash;
    THWMVMQ = IHWMVMQ + cdash;

    TLWHVMQ = ILWHVMQ + cdash;
    TMWHVMQ = IMWHVMQ + cdash;
    THWHVMQ = IHWHVMQ + cdash;

    TLWLVHQ = ILWLVHQ + cdash;
    TMWLVHQ = IMWLVHQ + cdash;
    THWLVHQ = IHWLVHQ + cdash;

    TLWMVHQ = ILWMVHQ + cdash;
    TMWMVHQ = IMWMVHQ + cdash;
    THWMVHQ = IHWMVHQ + cdash;

    TLWHVHQ = ILWHVHQ + cdash;
    TMWHVHQ = IMWHVHQ + cdash;
    THWHVHQ = IHWHVHQ + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
    PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
    PLWHVMQ PMWHVMQ PHWHVMQ
```

```
    PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
    PLWHVHQ PMWHVHQ PHWHVHQ);

    LOOP(XVAL,1,5,0.1);

    PLWLVLQ = ILWLVLQ*XVAL;
    PMWLVLQ = IMWLVLQ*XVAL;
    PHWLVLQ = IHWLVLQ*XVAL;

    PLWMVLQ = ILWMVLQ*XVAL;
    PMWMVLQ = IMWMVLQ*XVAL;
    PHWMVLQ = IHWMVLQ*XVAL;

    PLWHVLQ = ILWHVLQ*XVAL;
    PMWHVLQ = IMWHVLQ*XVAL;
    PHWHVLQ = IHWHVLQ*XVAL;

    PLWLVMQ = ILWLVMQ*XVAL;
    PMWLVMQ = IMWLVMQ*XVAL;
    PHWLVMQ = IHWLVMQ*XVAL;

    PLWMVMQ = ILWMVMQ*XVAL;
    PMWMVMQ = IMWMVMQ*XVAL;
    PHWMVMQ = IHWMVMQ*XVAL;

    PLWHVMQ = ILWHVMQ*XVAL;
    PMWHVMQ = IMWHVMQ*XVAL;
    PHWHVMQ = IHWHVMQ*XVAL;

    PLWLVHQ = ILWLVHQ*XVAL;
    PMWLVHQ = IMWLVHQ*XVAL;
    PHWLVHQ = IHWLVHQ*XVAL;

    PLWMVHQ = ILWMVHQ*XVAL;
    PMWMVHQ = IMWMVHQ*XVAL;
    PHWMVHQ = IHWMVHQ*XVAL;

    PLWHVHQ = ILWHVHQ*XVAL;
    PMWHVHQ = IMWHVHQ*XVAL;
    PHWHVHQ = IHWHVHQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 36: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating both the Mediator-DV path and the IV-DV path

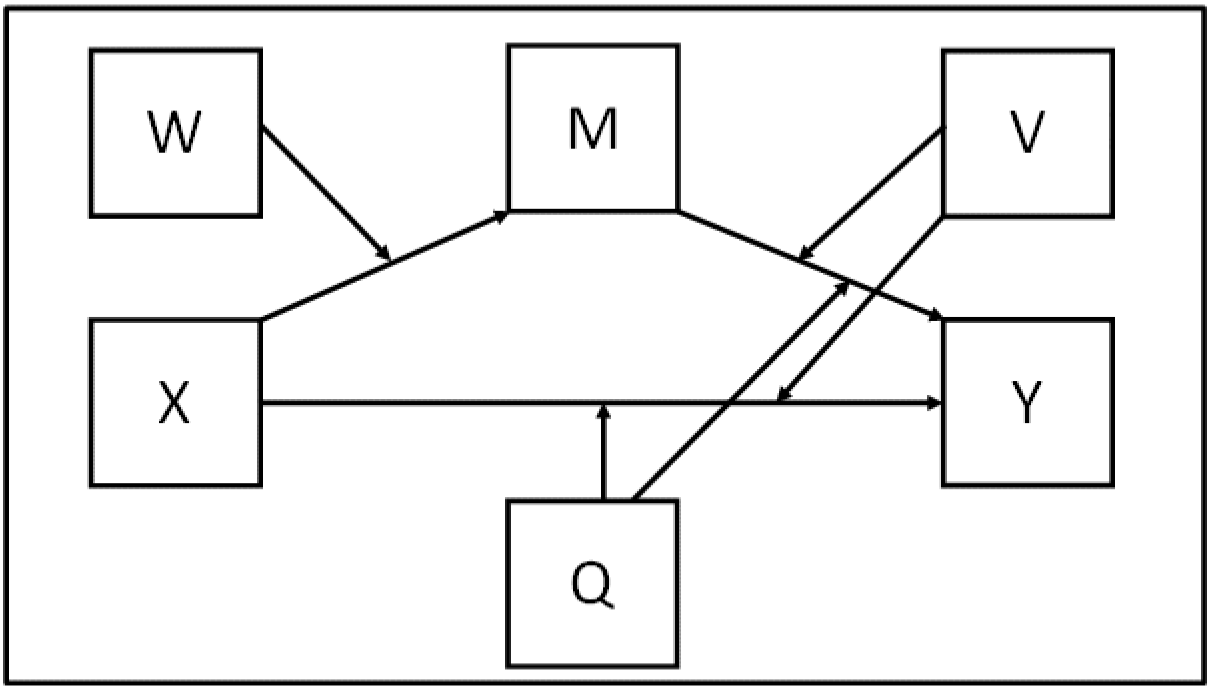
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

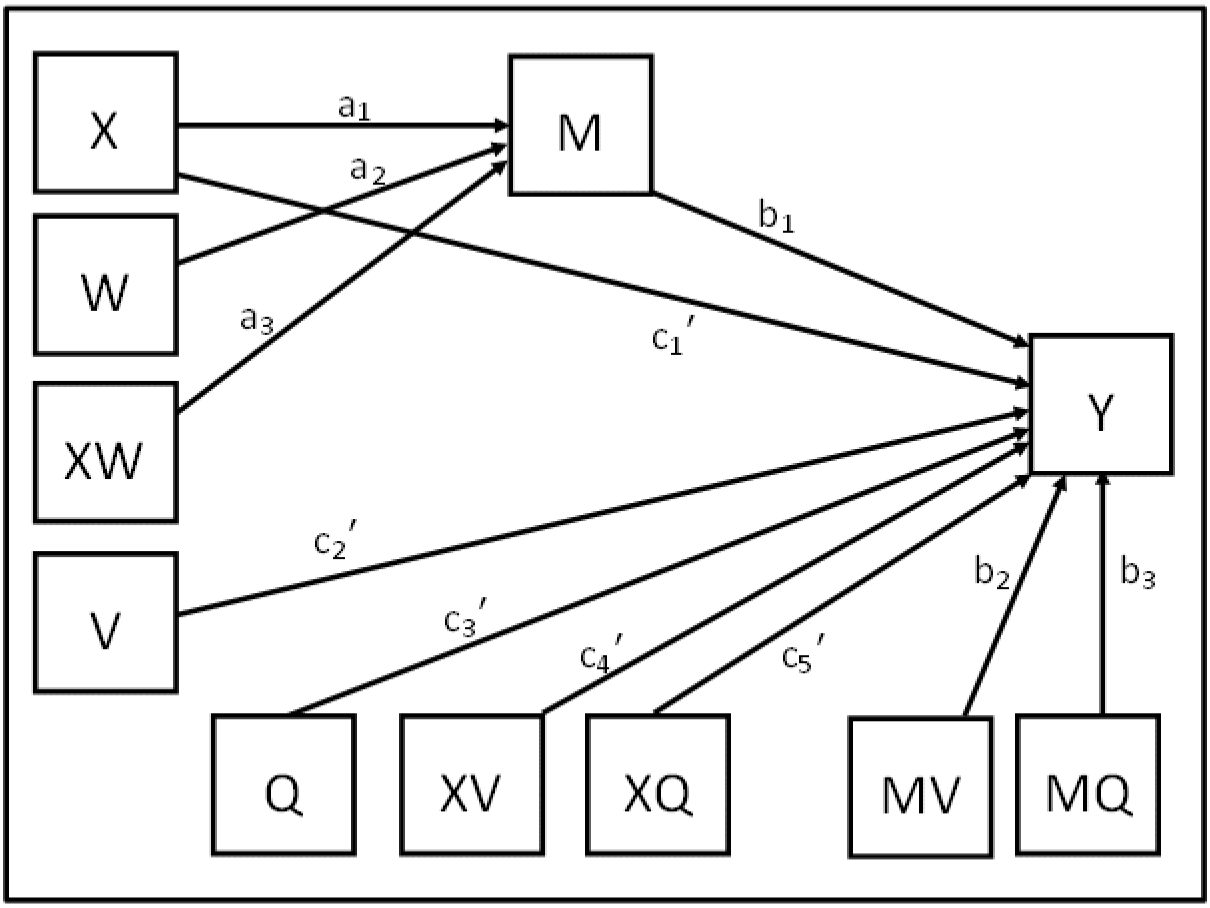
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2(a0 + a1X + a2W + a3XW)V + b3(a0 + a1X + a2W + a3XW)Q + c1'X + c2'V + c3'Q + c4'XV + c5'XQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + a0b2V + a1b2XV + a2b2WV + a3b2XWV + a0b3 + a1b3XQ + a2b3WQ + a3b3XWQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a0b2V + a2b2WV + a0b3 + a2b3WQ + c2'V + c3'Q) + (a1b1 + a3b1W + a1b2V + a3b2WV + a1b3Q + a3b3WQ + c1' + c4'V + c5'Q)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V, Q:

a1b1 + a3b1W + a1b2V + a3b2WV + a1b3Q + a3b3WQ = (a1 + a3W) (b1 + b2V + b3Q)

One direct effect of X on Y, conditional on V, Q:

c1' + c4'V + c5'Q

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW XV XQ MV MQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XQ = X*Q;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
```

```
    Y ON M (b1);
    Y ON MV (b2);
    Y ON MQ (b3);

    Y ON X (cdash1);
    Y ON V (cdash2);
    Y ON Q (cdash3);
    Y ON XV (cdash4);
    Y ON XQ (cdash5);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
    ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
    ILWHVLQ IMWHVLQ IHWHVLQ
    ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
    ILWHVMQ IMWHVMQ IHWHVMQ
    ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
    ILWHVHQ IMWHVHQ IHWHVHQ
    DLOV_LOQ DMEV_LOQ DHIV_LOQ DLOV_MEQ DMEV_MEQ DHIV_MEQ
    DLOV_HIQ DMEV_HIQ DHIV_HIQ
    TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
    TLWHVLQ TMWHVLQ THWHVLQ
    TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
    TLWHVMQ TMWHVMQ THWHVMQ
    TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
    TLWHVHQ TMWHVHQ THWHVHQ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
```

```
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q;

    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q;

    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q;
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q;
```

```
    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q;

    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q;
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q;

    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q;

    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q;
```

```
    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q;
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOV_LOQ = cdash1 + cdash4*LOW_V + cdash5*LOW_Q;
    DMEV_LOQ = cdash1 + cdash4*MED_W + cdash5*LOW_Q;
    DHIV_LOQ = cdash1 + cdash4*HIGH_V + cdash5*LOW_Q;

    DLOV_MEQ = cdash1 + cdash4*LOW_V + cdash5*MED_Q;
    DMEV_MEQ = cdash1 + cdash4*MED_W + cdash5*MED_Q;
    DHIV_MEQ = cdash1 + cdash4*HIGH_V + cdash5*MED_Q;

    DLOV_HIQ = cdash1 + cdash4*LOW_V + cdash5*HIGH_Q;
    DMEV_HIQ = cdash1 + cdash4*MED_W + cdash5*HIGH_Q;
    DHIV_HIQ = cdash1 + cdash4*HIGH_V + cdash5*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLWLVLQ = ILWLVLQ + DLOV_LOQ;
    TMWLVLQ = IMWLVLQ + DLOV_LOQ;
    THWLVLQ = IHWLVLQ + DLOV_LOQ;

    TLWMVLQ = ILWMVLQ + DMEV_LOQ;
    TMWMVLQ = IMWMVLQ + DMEV_LOQ;
    THWMVLQ = IHWMVLQ + DMEV_LOQ;

    TLWHVLQ = ILWHVLQ + DHIV_LOQ;
    TMWHVLQ = IMWHVLQ + DHIV_LOQ;
    THWHVLQ = IHWHVLQ + DHIV_LOQ;

    TLWLVMQ = ILWLVMQ + DLOV_MEQ;
    TMWLVMQ = IMWLVMQ + DLOV_MEQ;
    THWLVMQ = IHWLVMQ + DLOV_MEQ;

    TLWMVMQ = ILWMVMQ + DMEV_MEQ;
    TMWMVMQ = IMWMVMQ + DMEV_MEQ;
    THWMVMQ = IHWMVMQ + DMEV_MEQ;

    TLWHVMQ = ILWHVMQ + DHIV_MEQ;
    TMWHVMQ = IMWHVMQ + DHIV_MEQ;
    THWHVMQ = IHWHVMQ + DHIV_MEQ;
```

```
    TLWLVHQ = ILWLVHQ + DLOV_HIQ;
    TMWLVHQ = IMWLVHQ + DLOV_HIQ;
    THWLVHQ = IHWLVHQ + DLOV_HIQ;

    TLWMVHQ = ILWMVHQ + DMEV_HIQ;
    TMWMVHQ = IMWMVHQ + DMEV_HIQ;
    THWMVHQ = IHWMVHQ + DMEV_HIQ;

    TLWHVHQ = ILWHVHQ + DHIV_HIQ;
    TMWHVHQ = IMWHVHQ + DHIV_HIQ;
    THWHVHQ = IHWHVHQ + DHIV_HIQ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
    PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
    PLWHVMQ PMWHVMQ PHWHVMQ
    PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
    PLWHVHQ PMWHVHQ PHWHVHQ);

    LOOP(XVAL,1,5,0.1);

    PLWLVLQ = ILWLVLQ*XVAL;
    PMWLVLQ = IMWLVLQ*XVAL;
    PHWLVLQ = IHWLVLQ*XVAL;

    PLWMVLQ = ILWMVLQ*XVAL;
    PMWMVLQ = IMWMVLQ*XVAL;
    PHWMVLQ = IHWMVLQ*XVAL;

    PLWHVLQ = ILWHVLQ*XVAL;
    PMWHVLQ = IMWHVLQ*XVAL;
    PHWHVLQ = IHWHVLQ*XVAL;

    PLWLVMQ = ILWLVMQ*XVAL;
    PMWLVMQ = IMWLVMQ*XVAL;
    PHWLVMQ = IHWLVMQ*XVAL;

    PLWMVMQ = ILWMVMQ*XVAL;
    PMWMVMQ = IMWMVMQ*XVAL;
    PHWMVMQ = IHWMVMQ*XVAL;

    PLWHVMQ = ILWHVMQ*XVAL;
    PMWHVMQ = IMWHVMQ*XVAL;
    PHWHVMQ = IHWHVMQ*XVAL;
```

```
      PLWLVHQ = ILWLVHQ*XVAL;
      PMWLVHQ = IMWLVHQ*XVAL;
      PHWLVHQ = IHWLVHQ*XVAL;

      PLWMVHQ = ILWMVHQ*XVAL;
      PMWMVHQ = IMWMVHQ*XVAL;
      PHWMVHQ = IHWMVHQ*XVAL;

      PLWHVHQ = ILWHVHQ*XVAL;
      PMWHVHQ = IMWHVHQ*XVAL;
      PHWHVHQ = IHWHVHQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 37: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating the Mediator-DV path with all 2-way and 3-way interactions

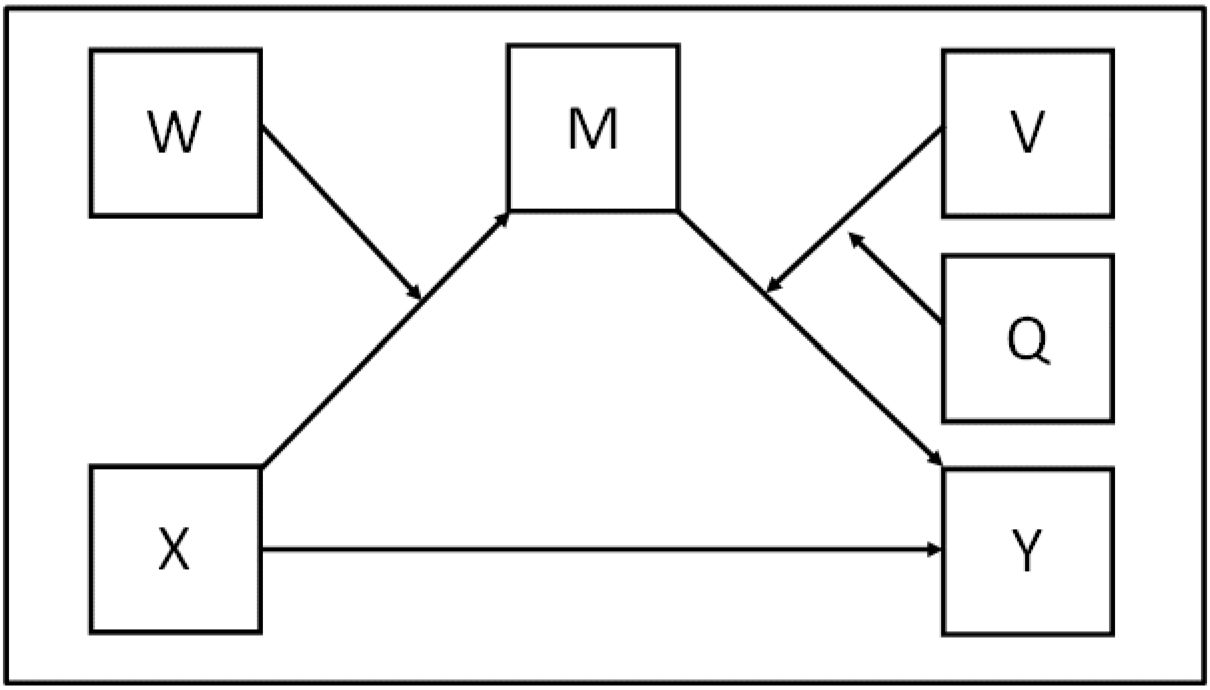
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

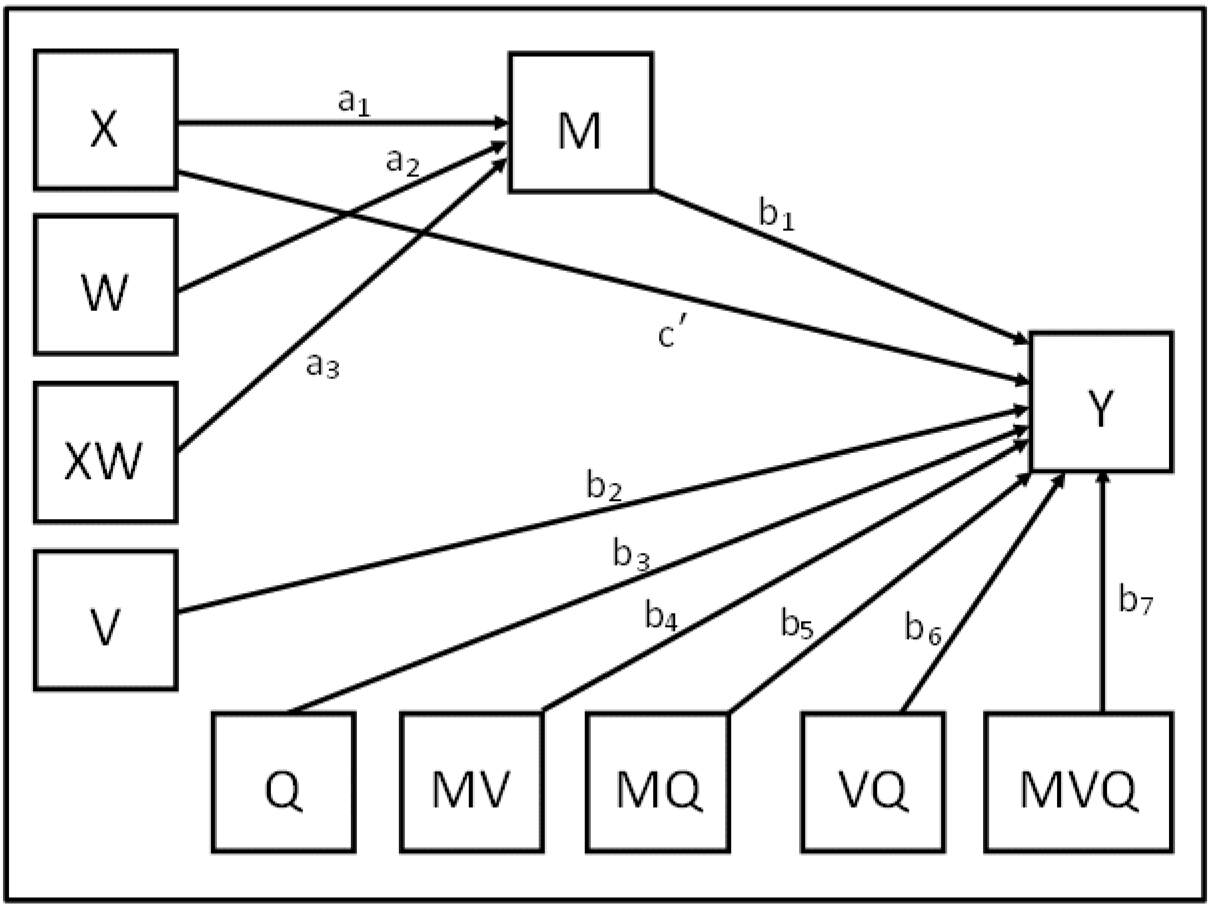
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c'X
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c'X
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2V + b3Q + b4(a0 + a1X + a2W + a3XW)V + b5(a0 + a1X + a2W + a3XW)Q + b6VQ + b7(a0 + a1X + a2W + a3XW)VQ + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4XWV + a0b5Q + a1b5XQ + a2b5WQ + a3b5XWQ + b6VQ + a0b7VQ + a1b7XVQ + a2b7WVQ + a3b7XWVQ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2V + b3Q + a0b4V + a2b4WV + a0b5Q + a2b5WQ + b6VQ + a0b7VQ + a2b7WVQ) + (a1b1 + a3b1W + a1b4V + a3b4WV + a1b5Q + a3b5WQ + a1b7VQ + a3b7WVQ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, V, Q:

a1b1 + a3b1W + a1b4V + a3b4WV + a1b5Q + a3b5WQ + a1b7VQ + a3b7WVQ = (a1 + a3W)(b1 + b4V + b5Q + b7VQ)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW VQ MV MQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   VQ = V*Q;
   MVQ = M*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON V (b2);
    Y ON Q (b3);
    Y ON MV (b4);
    Y ON MQ (b5);
    Y ON VQ (b6);
    Y ON MVQ (b7);

    Y ON X (cdash);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
     NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
     ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
     ILWHVLQ IMWHVLQ IHWHVLQ
     ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
     ILWHVMQ IMWHVMQ IHWHVMQ
     ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
     ILWHVHQ IMWHVHQ IHWHVHQ
     TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
     TLWHVLQ TMWHVLQ THWHVLQ
     TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
     TLWHVMQ TMWHVMQ THWHVMQ
     TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
     TLWHVHQ TMWHVHQ THWHVHQ);

     LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
     MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
```

```
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q + a1*b7*LOW_V*LOW_Q +
     a3*b7*LOW_W*LOW_V*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q + a1*b7*LOW_V*LOW_Q +
     a3*b7*MED_W*LOW_V*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q + a1*b7*LOW_V*LOW_Q +
     a3*b7*HIGH_W*LOW_V*LOW_Q;

    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q + a1*b7*MED_V*LOW_Q +
     a3*b7*LOW_W*MED_V*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q + a1*b7*MED_V*LOW_Q +
     a3*b7*MED_W*MED_V*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q + a1*b7*MED_V*LOW_Q +
     a3*b7*HIGH_W*MED_V*LOW_Q;

    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q + a1*b7*HIGH_V*LOW_Q +
     a3*b7*LOW_W*HIGH_V*LOW_Q;
```

```
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q + a1*b7*HIGH_V*LOW_Q +
     a3*b7*MED_W*HIGH_V*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q + a1*b7*HIGH_V*LOW_Q +
     a3*b7*HIGH_W*HIGH_V*LOW_Q;

    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q + a1*b7*LOW_V*MED_Q +
     a3*b7*LOW_W*LOW_V*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q + a1*b7*LOW_V*MED_Q +
     a3*b7*MED_W*LOW_V*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q + a1*b7*LOW_V*MED_Q +
     a3*b7*HIGH_W*LOW_V*MED_Q;

    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q + a1*b7*MED_V*MED_Q +
     a3*b7*LOW_W*MED_V*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q + a1*b7*MED_V*MED_Q +
     a3*b7*MED_W*MED_V*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q + a1*b7*MED_V*MED_Q +
     a3*b7*HIGH_W*MED_V*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q + a1*b7*HIGH_V*MED_Q +
     a3*b7*LOW_W*HIGH_V*MED_Q;
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q + a1*b7*HIGH_V*MED_Q +
     a3*b7*MED_W*HIGH_V*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q + a1*b7*HIGH_V*MED_Q +
     a3*b7*HIGH_W*HIGH_V*MED_Q;
```

```
    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q + a1*b7*LOW_V*HIGH_Q +
     a3*b7*LOW_W*LOW_V*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q + a1*b7*LOW_V*HIGH_Q +
     a3*b7*MED_W*LOW_V*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q + a1*b7*LOW_V*HIGH_Q
+
     a3*b7*HIGH_W*LOW_V*HIGH_Q;

    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q + a1*b7*MED_V*HIGH_Q +
     a3*b7*LOW_W*MED_V*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q + a1*b7*MED_V*HIGH_Q +
     a3*b7*MED_W*MED_V*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q + a1*b7*MED_V*HIGH_Q
+
     a3*b7*HIGH_W*MED_V*HIGH_Q;

    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q + a1*b7*HIGH_V*HIGH_Q
+
     a3*b7*LOW_W*HIGH_V*HIGH_Q;
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q + a1*b7*HIGH_V*HIGH_Q
+
     a3*b7*MED_W*HIGH_V*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q + a1*b7*HIGH_V*HIGH_Q
+
     a3*b7*HIGH_W*HIGH_V*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values
```

```
    TLWLVLQ = ILWLVLQ + cdash;
    TMWLVLQ = IMWLVLQ + cdash;
    THWLVLQ = IHWLVLQ + cdash;

    TLWMVLQ = ILWMVLQ + cdash;
    TMWMVLQ = IMWMVLQ + cdash;
    THWMVLQ = IHWMVLQ + cdash;

    TLWHVLQ = ILWHVLQ + cdash;
    TMWHVLQ = IMWHVLQ + cdash;
    THWHVLQ = IHWHVLQ + cdash;

    TLWLVMQ = ILWLVMQ + cdash;
    TMWLVMQ = IMWLVMQ + cdash;
    THWLVMQ = IHWLVMQ + cdash;

    TLWMVMQ = ILWMVMQ + cdash;
    TMWMVMQ = IMWMVMQ + cdash;
    THWMVMQ = IHWMVMQ + cdash;

    TLWHVMQ = ILWHVMQ + cdash;
    TMWHVMQ = IMWHVMQ + cdash;
    THWHVMQ = IHWHVMQ + cdash;

    TLWLVHQ = ILWLVHQ + cdash;
    TMWLVHQ = IMWLVHQ + cdash;
    THWLVHQ = IHWLVHQ + cdash;

    TLWMVHQ = ILWMVHQ + cdash;
    TMWMVHQ = IMWMVHQ + cdash;
    THWMVHQ = IHWMVHQ + cdash;

    TLWHVHQ = ILWHVHQ + cdash;
    TMWHVHQ = IMWHVHQ + cdash;
    THWHVHQ = IHWHVHQ + cdash;
! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis
    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
    PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
    PLWHVMQ PMWHVMQ PHWHVMQ
    PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
    PLWHVHQ PMWHVHQ PHWHVHQ);

    LOOP(XVAL,1,5,0.1);
```

```
    PLWLVLQ = ILWLVLQ*XVAL;
    PMWLVLQ = IMWLVLQ*XVAL;
    PHWLVLQ = IHWLVLQ*XVAL;

    PLWMVLQ = ILWMVLQ*XVAL;
    PMWMVLQ = IMWMVLQ*XVAL;
    PHWMVLQ = IHWMVLQ*XVAL;

    PLWHVLQ = ILWHVLQ*XVAL;
    PMWHVLQ = IMWHVLQ*XVAL;
    PHWHVLQ = IHWHVLQ*XVAL;

    PLWLVMQ = ILWLVMQ*XVAL;
    PMWLVMQ = IMWLVMQ*XVAL;
    PHWLVMQ = IHWLVMQ*XVAL;

    PLWMVMQ = ILWMVMQ*XVAL;
    PMWMVMQ = IMWMVMQ*XVAL;
    PHWMVMQ = IHWMVMQ*XVAL;

    PLWHVMQ = ILWHVMQ*XVAL;
    PMWHVMQ = IMWHVMQ*XVAL;
    PHWHVMQ = IHWHVMQ*XVAL;

    PLWLVHQ = ILWLVHQ*XVAL;
    PMWLVHQ = IMWLVHQ*XVAL;
    PHWLVHQ = IHWLVHQ*XVAL;

    PLWMVHQ = ILWMVHQ*XVAL;
    PMWMVHQ = IMWMVHQ*XVAL;
    PHWMVHQ = IHWMVHQ*XVAL;

    PLWHVHQ = ILWHVHQ*XVAL;
    PMWHVHQ = IMWHVHQ*XVAL;
    PHWHVHQ = IHWHVHQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 38: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating both the Mediator-DV path and the IV-DV path, with all 2-way and 3-way interactions

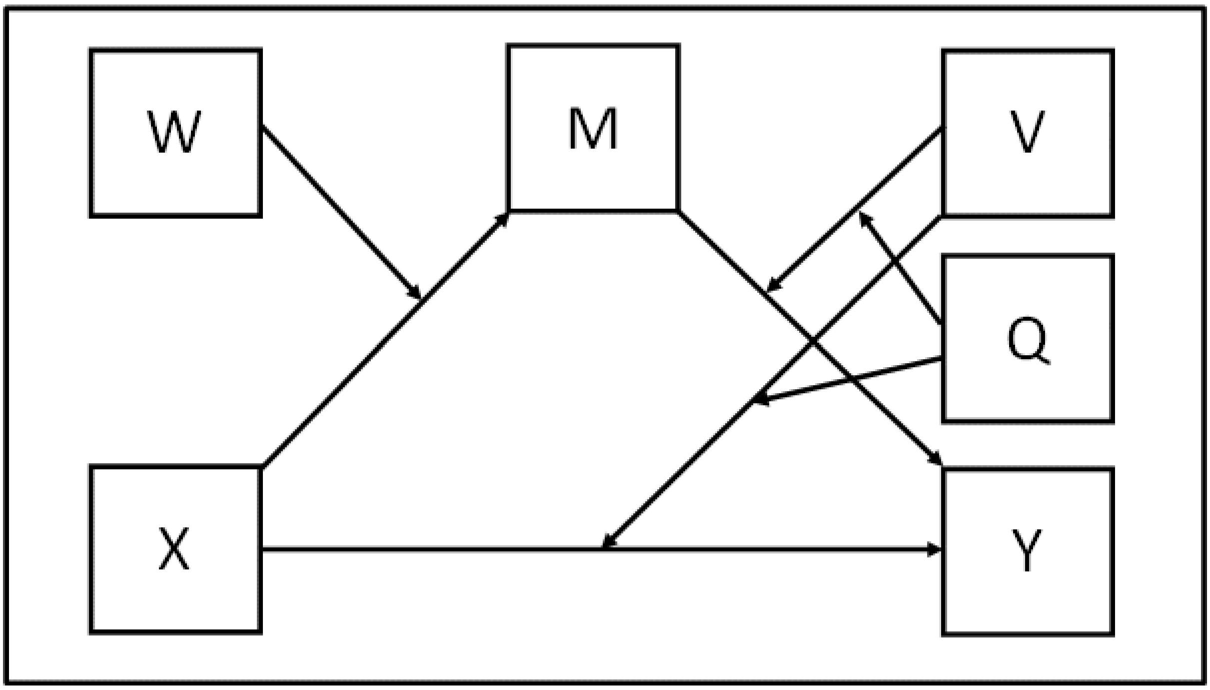
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

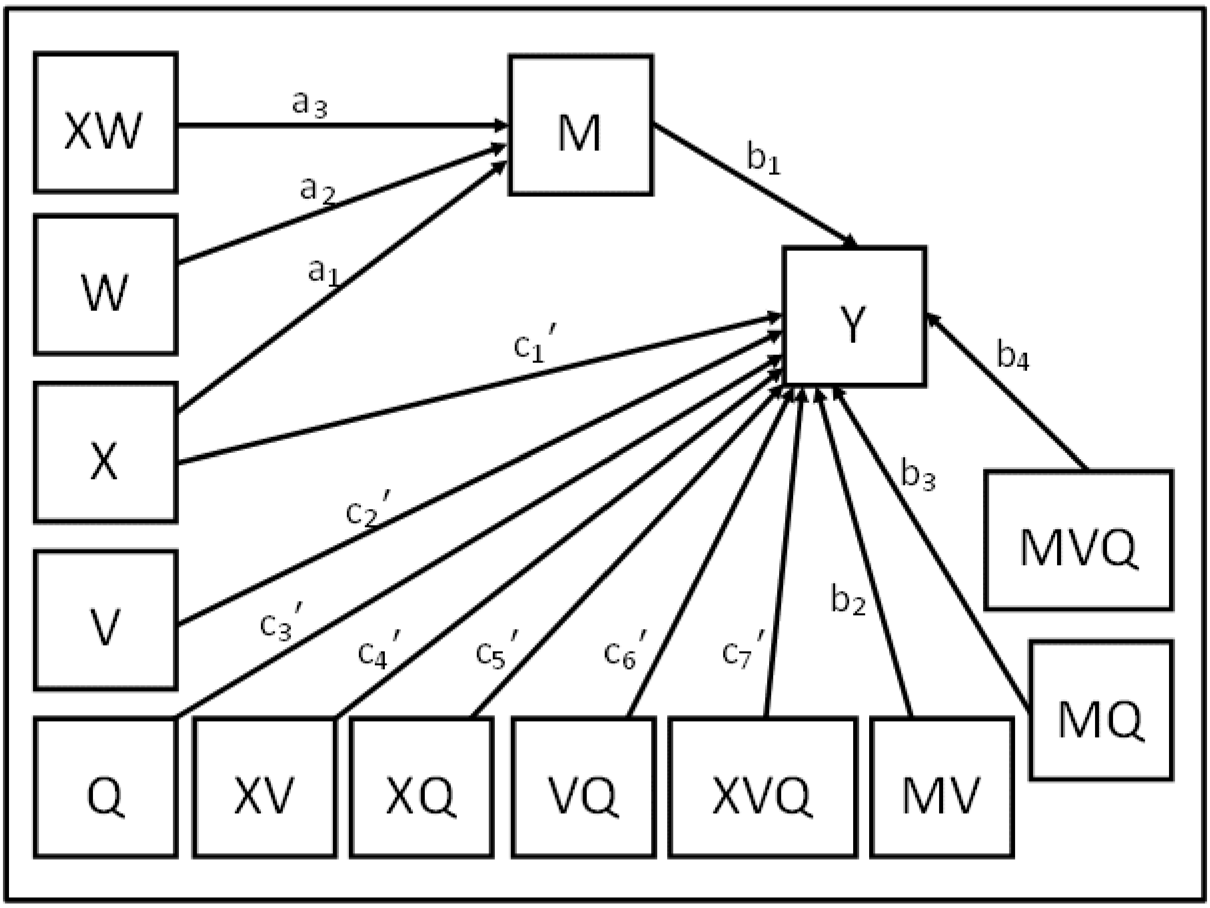
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2(a0 + a1X + a2W + a3XW)V + b3(a0 + a1X + a2W + a3XW)Q + b4(a0 + a1X + a2W + a3XW)VQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + a0b2V + a1b2XV + a2b2WV + a3b2XWV + a0b3Q + a1b3XQ + a2b3WQ + a3b3XWQ + a0b4VQ + a1b4XVQ + a2b4WVQ + a3b4XWVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a0b2V + a2b2WV + a0b3Q + a2b3WQ + a0b4VQ + a2b4WVQ + c2'V + c3'Q + c6'VQ) + (a1b1 + a3b1W + a1b2V + a3b2WV + a1b3Q + a3b3WQ + a1b4VQ + a3b4WVQ + c1' + c4'V + c5'Q + c7'VQ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V, Q:

a1b1 + a3b1W + a1b2V + a3b2WV + a1b3Q + a3b3WQ + a1b4VQ + a3b4WVQ = (a1 + a3W)(b1 + b2V + b3Q + b4VQ)

One direct effect of X on Y, conditional on V, Q:

c1' + c4'V + c5'Q + c7'VQ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW XV XQ VQ MV MQ XVQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XQ = X*Q;
   XV = X*V;
   VQ = V*Q;
   MVQ = M*V*Q;
   XVQ = X*V*Q;

ANALYSIS:
   TYPE = GENERAL;
```

```
    ESTIMATOR = ML;
    BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON MV (b2);
    Y ON MQ (b3);
    Y ON MVQ (b4);

    Y ON X (cdash1);
    Y ON V (cdash2);
    Y ON Q (cdash3);
    Y ON XV (cdash4);
    Y ON XQ (cdash5);
    Y ON VQ (cdash6);
    Y ON XVQ (cdash7);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
     NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
     ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
     ILWHVLQ IMWHVLQ IHWHVLQ
     ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
     ILWHVMQ IMWHVMQ IHWHVMQ
     ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
     ILWHVHQ IMWHVHQ IHWHVHQ
     DLOV_LOQ DMEV_LOQ DHIV_LOQ DLOV_MEQ DMEV_MEQ DHIV_MEQ
     DLOV_HIQ DMEV_HIQ DHIV_HIQ
     TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
```

```
    TLWHVLQ TMWHVLQ THWHVLQ
    TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
    TLWHVMQ TMWHVMQ THWHVMQ
    TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
    TLWHVHQ TMWHVHQ THWHVHQ);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q + a1*b4*LOW_V*LOW_Q +
     a3*b4*LOW_W*LOW_V*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q + a1*b4*LOW_V*LOW_Q +
     a3*b4*MED_W*LOW_V*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q + a1*b4*LOW_V*LOW_Q +
     a3*b4*HIGH_W*LOW_V*LOW_Q;

    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q + a1*b4*MED_V*LOW_Q +
     a3*b4*LOW_W*MED_V*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q + a1*b4*MED_V*LOW_Q +
```

```
     a3*b4*MED_W*MED_V*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q + a1*b4*MED_V*LOW_Q +
     a3*b4*HIGH_W*MED_V*LOW_Q;

    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q + a1*b4*HIGH_V*LOW_Q +
     a3*b4*LOW_W*HIGH_V*LOW_Q;
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q + a1*b4*HIGH_V*LOW_Q +
     a3*b4*MED_W*HIGH_V*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q + a1*b4*HIGH_V*LOW_Q +
     a3*b4*HIGH_W*HIGH_V*LOW_Q;

    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q + a1*b4*LOW_V*MED_Q +
     a3*b4*LOW_W*LOW_V*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q + a1*b4*LOW_V*MED_Q +
     a3*b4*MED_W*LOW_V*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q + a1*b4*LOW_V*MED_Q +
     a3*b4*HIGH_W*LOW_V*MED_Q;

    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q + a1*b4*MED_V*MED_Q +
     a3*b4*LOW_W*MED_V*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q + a1*b4*MED_V*MED_Q +
     a3*b4*MED_W*MED_V*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q + a1*b4*MED_V*MED_Q +
     a3*b4*HIGH_W*MED_V*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q + a1*b4*HIGH_V*MED_Q +
     a3*b4*LOW_W*HIGH_V*MED_Q;
```

```
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q + a1*b4*HIGH_V*MED_Q +
     a3*b4*MED_W*HIGH_V*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q + a1*b4*HIGH_V*MED_Q +
     a3*b4*HIGH_W*HIGH_V*MED_Q;

    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q + a1*b4*LOW_V*HIGH_Q +
     a3*b4*LOW_W*LOW_V*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q + a1*b4*LOW_V*HIGH_Q +
     a3*b4*MED_W*LOW_V*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q + a1*b4*LOW_V*HIGH_Q
+
     a3*b4*HIGH_W*LOW_V*HIGH_Q;

    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q + a1*b4*MED_V*HIGH_Q +
     a3*b4*LOW_W*MED_V*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q + a1*b4*MED_V*HIGH_Q +
     a3*b4*MED_W*MED_V*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q + a1*b4*MED_V*HIGH_Q
+
     a3*b4*HIGH_W*MED_V*HIGH_Q;

    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q + a1*b4*HIGH_V*HIGH_Q
+
     a3*b4*LOW_W*HIGH_V*HIGH_Q;
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q + a1*b4*HIGH_V*HIGH_Q
+
     a3*b4*MED_W*HIGH_V*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
```

```
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q + a1*b4*HIGH_V*HIGH_Q
+
     a3*b4*HIGH_W*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOV_LOQ = cdash1 + cdash4*LOW_V + cdash5*LOW_Q +
cdash7*LOW_V*LOW_Q;
    DMEV_LOQ = cdash1 + cdash4*MED_W + cdash5*LOW_Q +
cdash7*MED_W*LOW_Q;
    DHIV_LOQ = cdash1 + cdash4*HIGH_V + cdash5*LOW_Q +
cdash7*HIGH_V*LOW_Q;

    DLOV_MEQ = cdash1 + cdash4*LOW_V + cdash5*MED_Q +
cdash7*LOW_V*MED_Q;
    DMEV_MEQ = cdash1 + cdash4*MED_W + cdash5*MED_Q +
cdash7*MED_W*MED_Q;
    DHIV_MEQ = cdash1 + cdash4*HIGH_V + cdash5*MED_Q +
cdash7*HIGH_V*MED_Q;

    DLOV_HIQ = cdash1 + cdash4*LOW_V + cdash5*HIGH_Q +
cdash7*LOW_V*HIGH_Q;
    DMEV_HIQ = cdash1 + cdash4*MED_W + cdash5*HIGH_Q +
cdash7*MED_W*HIGH_Q;
    DHIV_HIQ = cdash1 + cdash4*HIGH_V + cdash5*HIGH_Q +
cdash7*HIGH_V*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLWLVLQ = ILWLVLQ + DLOV_LOQ;
    TMWLVLQ = IMWLVLQ + DLOV_LOQ;
    THWLVLQ = IHWLVLQ + DLOV_LOQ;

    TLWMVLQ = ILWMVLQ + DMEV_LOQ;
    TMWMVLQ = IMWMVLQ + DMEV_LOQ;
    THWMVLQ = IHWMVLQ + DMEV_LOQ;

    TLWHVLQ = ILWHVLQ + DHIV_LOQ;
    TMWHVLQ = IMWHVLQ + DHIV_LOQ;
    THWHVLQ = IHWHVLQ + DHIV_LOQ;

    TLWLVMQ = ILWLVMQ + DLOV_MEQ;
    TMWLVMQ = IMWLVMQ + DLOV_MEQ;
    THWLVMQ = IHWLVMQ + DLOV_MEQ;

    TLWMVMQ = ILWMVMQ + DMEV_MEQ;
    TMWMVMQ = IMWMVMQ + DMEV_MEQ;
    THWMVMQ = IHWMVMQ + DMEV_MEQ;
```

```
    TLWHVMQ = ILWHVMQ + DHIV_MEQ;
    TMWHVMQ = IMWHVMQ + DHIV_MEQ;
    THWHVMQ = IHWHVMQ + DHIV_MEQ;

    TLWLVHQ = ILWLVHQ + DLOV_HIQ;
    TMWLVHQ = IMWLVHQ + DLOV_HIQ;
    THWLVHQ = IHWLVHQ + DLOV_HIQ;

    TLWMVHQ = ILWMVHQ + DMEV_HIQ;
    TMWMVHQ = IMWMVHQ + DMEV_HIQ;
    THWMVHQ = IHWMVHQ + DMEV_HIQ;

    TLWHVHQ = ILWHVHQ + DHIV_HIQ;
    TMWHVHQ = IMWHVHQ + DHIV_HIQ;
    THWHVHQ = IHWHVHQ + DHIV_HIQ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
    PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
    PLWHVMQ PMWHVMQ PHWHVMQ
    PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
    PLWHVHQ PMWHVHQ PHWHVHQ);

    LOOP(XVAL,1,5,0.1);

    PLWLVLQ = ILWLVLQ*XVAL;
    PMWLVLQ = IMWLVLQ*XVAL;
    PHWLVLQ = IHWLVLQ*XVAL;

    PLWMVLQ = ILWMVLQ*XVAL;
    PMWMVLQ = IMWMVLQ*XVAL;
    PHWMVLQ = IHWMVLQ*XVAL;

    PLWHVLQ = ILWHVLQ*XVAL;
    PMWHVLQ = IMWHVLQ*XVAL;
    PHWHVLQ = IHWHVLQ*XVAL;

    PLWLVMQ = ILWLVMQ*XVAL;
    PMWLVMQ = IMWLVMQ*XVAL;
    PHWLVMQ = IHWLVMQ*XVAL;

    PLWMVMQ = ILWMVMQ*XVAL;
    PMWMVMQ = IMWMVMQ*XVAL;
    PHWMVMQ = IHWMVMQ*XVAL;
```

```
    PLWHVMQ = ILWHVMQ*XVAL;
    PMWHVMQ = IMWHVMQ*XVAL;
    PHWHVMQ = IHWHVMQ*XVAL;

    PLWLVHQ = ILWLVHQ*XVAL;
    PMWLVHQ = IMWLVHQ*XVAL;
    PHWLVHQ = IHWLVHQ*XVAL;

    PLWMVHQ = ILWMVHQ*XVAL;
    PMWMVHQ = IMWMVHQ*XVAL;
    PHWMVHQ = IHWMVHQ*XVAL;

    PLWHVHQ = ILWHVHQ*XVAL;
    PMWHVHQ = IMWHVHQ*XVAL;
    PHWHVHQ = IHWHVHQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 39: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating the IV-Mediator path, 2 moderating the Mediator-DV path with all 2-way and 3-way interactions, 1 of which also moderates the direct IV-DV path

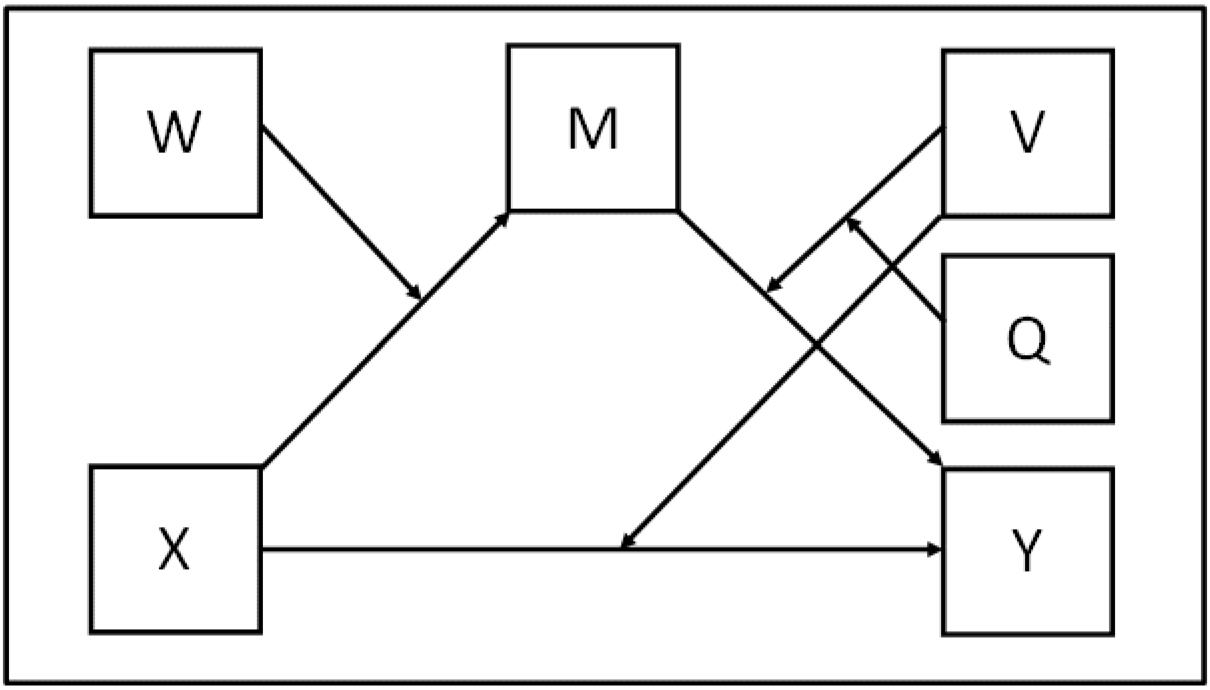
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

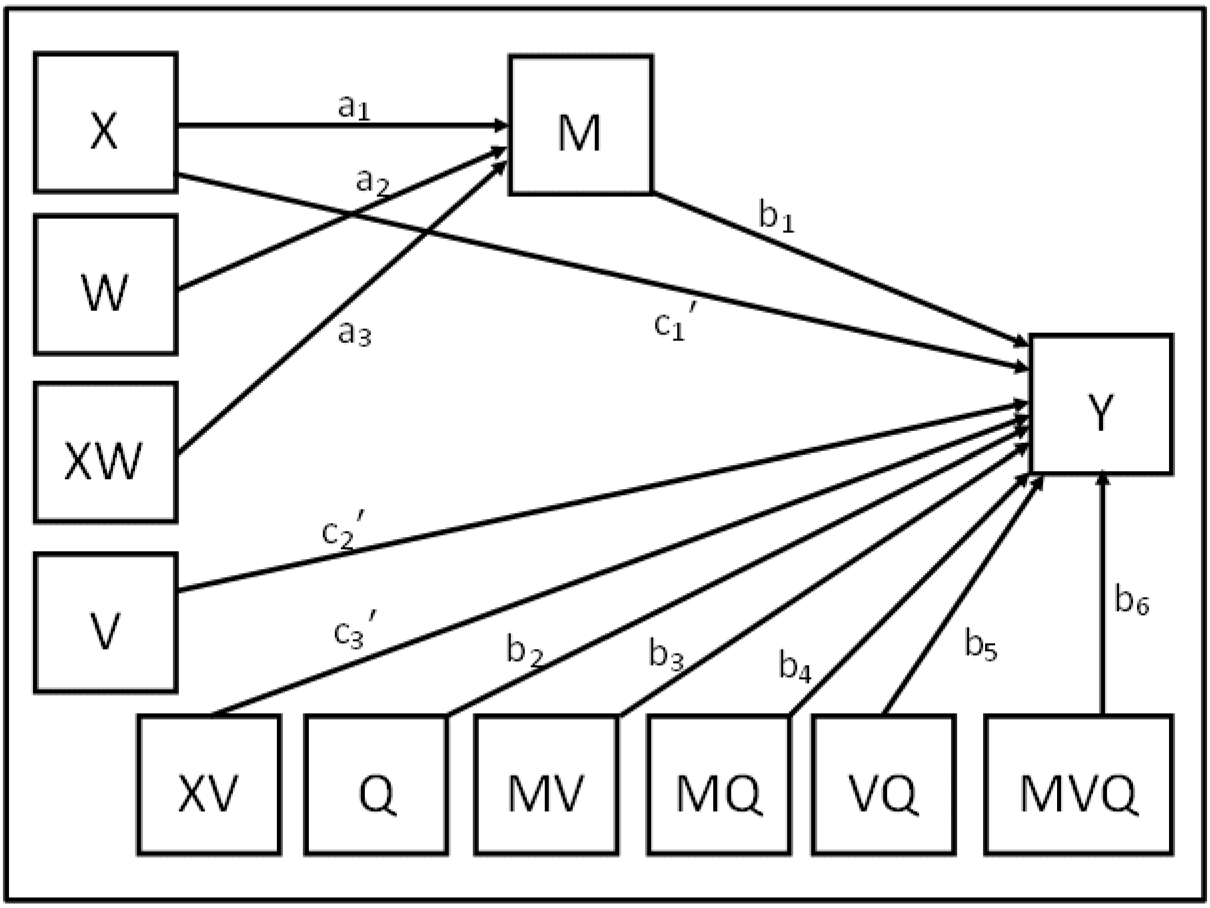
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2Q + b3MV + b4MQ + b5VQ + b6MVQ + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2Q + b3MV + b4MQ + b5VQ + b6MVQ + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2Q + b3(a0 + a1X + a2W + a3XW)V + b4(a0 + a1X + a2W + a3XW)Q + b5VQ + b6(a0 + a1X + a2W + a3XW)VQ + c1'X + c2'V + c3'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2Q + a0b3V + a1b3XV + a2b3WV + a3b3XWV + a0b4Q + a1b4XQ + a2b4WQ + a3b4XWQ + b5VQ + a0b6VQ + a1b6XVQ + a2b6WVQ + a3b6XWVQ + c1'X + c2'V + c3'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2Q + a0b3V + a2b3WV + a0b4Q + a2b4WQ + b5VQ + a0b6VQ + a2b6WVQ + c2'V) + (a1b1 + a3b1W + a1b3V + a3b3WV + a1b4Q + a3b4WQ + a1b6VQ + a3b6WVQ + c1' + c3'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V, Q:

a1b1 + a3b1W + a1b3V + a3b3WV + a1b4Q + a3b4WQ + a1b6VQ + a3b6WVQ = (a1 + a3W)(b1 + b3V + b4Q + b6VQ)

One direct effect of X on Y, conditional on V:

c1' + c3'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW XV VQ MV MQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XV = X*V;
   VQ = V*Q;
   MVQ = M*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON Q (b2);
   Y ON MV (b3);
   Y ON MQ (b4);
   Y ON VQ (b5);
   Y ON MVQ (b6);

   Y ON X (cdash1);
   Y ON V (cdash2);
   Y ON XV (cdash3);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
    ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
    ILWHVLQ IMWHVLQ IHWHVLQ
    ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
    ILWHVMQ IMWHVMQ IHWHVMQ
    ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
    ILWHVHQ IMWHVHQ IHWHVHQ
    DIR_LOWV DIR_MEDV DIR_HIV
    TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
    TLWHVLQ TMWHVLQ THWHVLQ
    TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
    TLWHVMQ TMWHVMQ THWHVMQ
    TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
    TLWHVHQ TMWHVHQ THWHVHQ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
```

```
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_V +
a3*b3*LOW_W*LOW_V +
     a1*b4*LOW_Q + a3*b4*LOW_W*LOW_Q + a1*b6*LOW_V*LOW_Q +
     a3*b6*LOW_W*LOW_V*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b3*LOW_V +
a3*b3*MED_W*LOW_V +
     a1*b4*LOW_Q + a3*b4*MED_W*LOW_Q + a1*b6*LOW_V*LOW_Q +
     a3*b6*MED_W*LOW_V*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b3*LOW_V +
a3*b3*HIGH_W*LOW_V +
     a1*b4*LOW_Q + a3*b4*HIGH_W*LOW_Q + a1*b6*LOW_V*LOW_Q +
     a3*b6*HIGH_W*LOW_V*LOW_Q;

    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b3*MED_V +
a3*b3*LOW_W*MED_V +
     a1*b4*LOW_Q + a3*b4*LOW_W*LOW_Q + a1*b6*MED_V*LOW_Q +
     a3*b6*LOW_W*MED_V*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b3*MED_V +
a3*b3*MED_W*MED_V +
     a1*b4*LOW_Q + a3*b4*MED_W*LOW_Q + a1*b6*MED_V*LOW_Q +
     a3*b6*MED_W*MED_V*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b3*MED_V +
a3*b3*HIGH_W*MED_V +
     a1*b4*LOW_Q + a3*b4*HIGH_W*LOW_Q + a1*b6*MED_V*LOW_Q +
     a3*b6*HIGH_W*MED_V*LOW_Q;

    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b3*HIGH_V +
a3*b3*LOW_W*HIGH_V +
     a1*b4*LOW_Q + a3*b4*LOW_W*LOW_Q + a1*b6*HIGH_V*LOW_Q +
```

```
     a3*b6*LOW_W*HIGH_V*LOW_Q;
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b3*HIGH_V +
a3*b3*MED_W*HIGH_V +
     a1*b4*LOW_Q + a3*b4*MED_W*LOW_Q + a1*b6*HIGH_V*LOW_Q +
     a3*b6*MED_W*HIGH_V*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_V +
a3*b3*HIGH_W*HIGH_V +
     a1*b4*LOW_Q + a3*b4*HIGH_W*LOW_Q + a1*b6*HIGH_V*LOW_Q +
     a3*b6*HIGH_W*HIGH_V*LOW_Q;

    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_V +
a3*b3*LOW_W*LOW_V +
     a1*b4*MED_Q + a3*b4*LOW_W*MED_Q + a1*b6*LOW_V*MED_Q +
     a3*b6*LOW_W*LOW_V*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b3*LOW_V +
a3*b3*MED_W*LOW_V +
     a1*b4*MED_Q + a3*b4*MED_W*MED_Q + a1*b6*LOW_V*MED_Q +
     a3*b6*MED_W*LOW_V*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b3*LOW_V +
a3*b3*HIGH_W*LOW_V +
     a1*b4*MED_Q + a3*b4*HIGH_W*MED_Q + a1*b6*LOW_V*MED_Q +
     a3*b6*HIGH_W*LOW_V*MED_Q;

    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b3*MED_V +
a3*b3*LOW_W*MED_V +
     a1*b4*MED_Q + a3*b4*LOW_W*MED_Q + a1*b6*MED_V*MED_Q +
     a3*b6*LOW_W*MED_V*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b3*MED_V +
a3*b3*MED_W*MED_V +
     a1*b4*MED_Q + a3*b4*MED_W*MED_Q + a1*b6*MED_V*MED_Q +
     a3*b6*MED_W*MED_V*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b3*MED_V +
a3*b3*HIGH_W*MED_V +
     a1*b4*MED_Q + a3*b4*HIGH_W*MED_Q + a1*b6*MED_V*MED_Q +
     a3*b6*HIGH_W*MED_V*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b3*HIGH_V +
a3*b3*LOW_W*HIGH_V +
     a1*b4*MED_Q + a3*b4*LOW_W*MED_Q + a1*b6*HIGH_V*MED_Q +
     a3*b6*LOW_W*HIGH_V*MED_Q;
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b3*HIGH_V +
a3*b3*MED_W*HIGH_V +
     a1*b4*MED_Q + a3*b4*MED_W*MED_Q + a1*b6*HIGH_V*MED_Q +
     a3*b6*MED_W*HIGH_V*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_V +
a3*b3*HIGH_W*HIGH_V +
     a1*b4*MED_Q + a3*b4*HIGH_W*MED_Q + a1*b6*HIGH_V*MED_Q +
     a3*b6*HIGH_W*HIGH_V*MED_Q;
```

```
    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_V +
a3*b3*LOW_W*LOW_V +
     a1*b4*HIGH_Q + a3*b4*LOW_W*HIGH_Q + a1*b6*LOW_V*HIGH_Q +
     a3*b6*LOW_W*LOW_V*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b3*LOW_V +
a3*b3*MED_W*LOW_V +
     a1*b4*HIGH_Q + a3*b4*MED_W*HIGH_Q + a1*b6*LOW_V*HIGH_Q +
     a3*b6*MED_W*LOW_V*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b3*LOW_V +
a3*b3*HIGH_W*LOW_V +
     a1*b4*HIGH_Q + a3*b4*HIGH_W*HIGH_Q + a1*b6*LOW_V*HIGH_Q
+
     a3*b6*HIGH_W*LOW_V*HIGH_Q;

    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b3*MED_V +
a3*b3*LOW_W*MED_V +
     a1*b4*HIGH_Q + a3*b4*LOW_W*HIGH_Q + a1*b6*MED_V*HIGH_Q +
     a3*b6*LOW_W*MED_V*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b3*MED_V +
a3*b3*MED_W*MED_V +
     a1*b4*HIGH_Q + a3*b4*MED_W*HIGH_Q + a1*b6*MED_V*HIGH_Q +
     a3*b6*MED_W*MED_V*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b3*MED_V +
a3*b3*HIGH_W*MED_V +
     a1*b4*HIGH_Q + a3*b4*HIGH_W*HIGH_Q + a1*b6*MED_V*HIGH_Q
+
     a3*b6*HIGH_W*MED_V*HIGH_Q;

    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b3*HIGH_V +
a3*b3*LOW_W*HIGH_V +
     a1*b4*HIGH_Q + a3*b4*LOW_W*HIGH_Q + a1*b6*HIGH_V*HIGH_Q
+
     a3*b6*LOW_W*HIGH_V*HIGH_Q;
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b3*HIGH_V +
a3*b3*MED_W*HIGH_V +
     a1*b4*HIGH_Q + a3*b4*MED_W*HIGH_Q + a1*b6*HIGH_V*HIGH_Q
+
     a3*b6*MED_W*HIGH_V*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_V +
a3*b3*HIGH_W*HIGH_V +
     a1*b4*HIGH_Q + a3*b4*HIGH_W*HIGH_Q + a1*b6*HIGH_V*HIGH_Q
+
     a3*b6*HIGH_W*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values
```

```
    DIR_LOWV = cdash1 + cdash3*LOW_V;
    DIR_MEDV = cdash1 + cdash3*MED_V;
    DIR_HIV = cdash1 + cdash3*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLWLVLQ = ILWLVLQ + DIR_LOWV;
    TMWLVLQ = IMWLVLQ + DIR_LOWV;
    THWLVLQ = IHWLVLQ + DIR_LOWV;

    TLWMVLQ = ILWMVLQ + DIR_MEDV;
    TMWMVLQ = IMWMVLQ + DIR_MEDV;
    THWMVLQ = IHWMVLQ + DIR_MEDV;

    TLWHVLQ = ILWHVLQ + DIR_HIV;
    TMWHVLQ = IMWHVLQ + DIR_HIV;
    THWHVLQ = IHWHVLQ + DIR_HIV;

    TLWLVMQ = ILWLVMQ + DIR_LOWV;
    TMWLVMQ = IMWLVMQ + DIR_LOWV;
    THWLVMQ = IHWLVMQ + DIR_LOWV;

    TLWMVMQ = ILWMVMQ + DIR_MEDV;
    TMWMVMQ = IMWMVMQ + DIR_MEDV;
    THWMVMQ = IHWMVMQ + DIR_MEDV;

    TLWHVMQ = ILWHVMQ + DIR_HIV;
    TMWHVMQ = IMWHVMQ + DIR_HIV;
    THWHVMQ = IHWHVMQ + DIR_HIV;

    TLWLVHQ = ILWLVHQ + DIR_LOWV;
    TMWLVHQ = IMWLVHQ + DIR_LOWV;
    THWLVHQ = IHWLVHQ + DIR_LOWV;

    TLWMVHQ = ILWMVHQ + DIR_MEDV;
    TMWMVHQ = IMWMVHQ + DIR_MEDV;
    THWMVHQ = IHWMVHQ + DIR_MEDV;

    TLWHVHQ = ILWHVHQ + DIR_HIV;
    TMWHVHQ = IMWHVHQ + DIR_HIV;
    THWHVHQ = IHWHVHQ + DIR_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
```

```
   PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
   PLWHVMQ PMWHVMQ PHWHVMQ
   PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
   PLWHVHQ PMWHVHQ PHWHVHQ);

   LOOP(XVAL,1,5,0.1);

   PLWLVLQ = ILWLVLQ*XVAL;
   PMWLVLQ = IMWLVLQ*XVAL;
   PHWLVLQ = IHWLVLQ*XVAL;

   PLWMVLQ = ILWMVLQ*XVAL;
   PMWMVLQ = IMWMVLQ*XVAL;
   PHWMVLQ = IHWMVLQ*XVAL;

   PLWHVLQ = ILWHVLQ*XVAL;
   PMWHVLQ = IMWHVLQ*XVAL;
   PHWHVLQ = IHWHVLQ*XVAL;

   PLWLVMQ = ILWLVMQ*XVAL;
   PMWLVMQ = IMWLVMQ*XVAL;
   PHWLVMQ = IHWLVMQ*XVAL;

   PLWMVMQ = ILWMVMQ*XVAL;
   PMWMVMQ = IMWMVMQ*XVAL;
   PHWMVMQ = IHWMVMQ*XVAL;

   PLWHVMQ = ILWHVMQ*XVAL;
   PMWHVMQ = IMWHVMQ*XVAL;
   PHWHVMQ = IHWHVMQ*XVAL;

   PLWLVHQ = ILWLVHQ*XVAL;
   PMWLVHQ = IMWLVHQ*XVAL;
   PHWLVHQ = IHWLVHQ*XVAL;

   PLWMVHQ = ILWMVHQ*XVAL;
   PMWMVHQ = IMWMVHQ*XVAL;
   PHWMVHQ = IHWMVHQ*XVAL;

   PLWHVHQ = ILWHVHQ*XVAL;
   PMWHVHQ = IMWHVHQ*XVAL;
   PHWHVHQ = IHWHVHQ*XVAL;

PLOT:
  TYPE = plot2;

OUTPUT:
  STAND CINT(bcbootstrap);
```

Model 40: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV- Mediator path and the direct IV-DV path, 2 moderating the Mediator-DV path

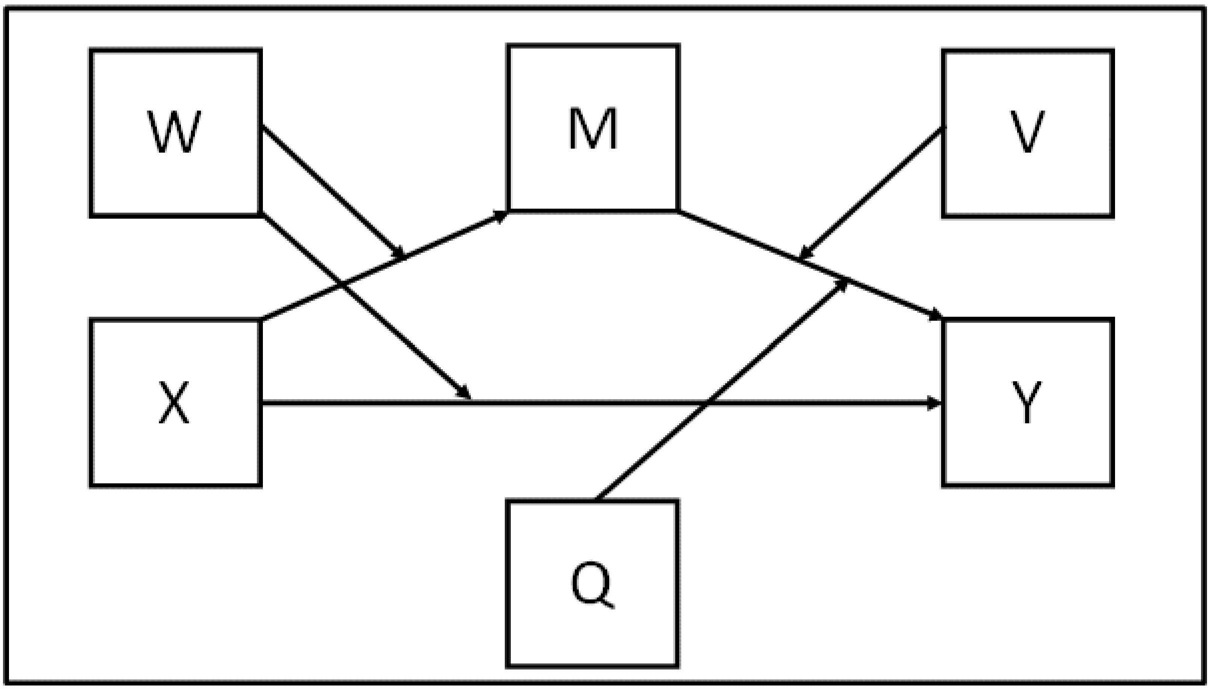
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

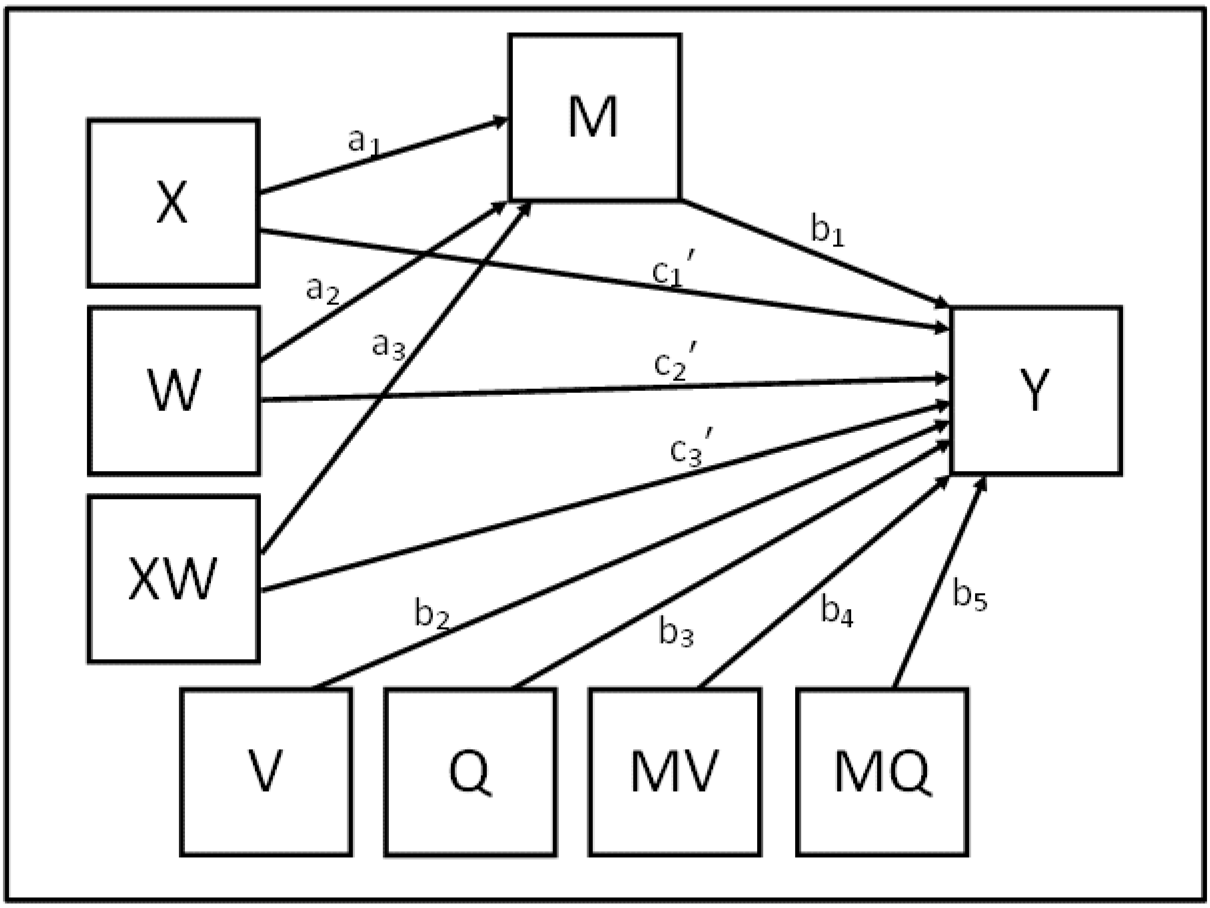
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2V + b3Q + b4(a0 + a1X + a2W + a3XW)V + b5(a0 + a1X + a2W + a3XW)Q + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4XWV + a0b5Q + a1b5XQ + a2b5WQ + a3b5XWQ + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX
Y = (b0 + a0b1 + a2b1W + b2V + b3Q + a0b4V + a2b4WV + a0b5Q + a2b5WQ + c2'W) + (a1b1 + a3b1W + a1b4V + a3b4WV + a1b5Q + a3b5WQ + c1' + c3'W)X

Hence...
One indirect effect(s) of X on Y, conditional on W, V, Q: a1b1 + a3b1W + a1b4V + a3b4WV + a1b5Q + a3b5WQ = (a1 + a3W)(b1 + b4V + b5Q)
One direct effect of X on Y, conditional on W: c1' + c3'W

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW MV MQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);
```

```
  [M] (a0);
  M ON X (a1);
  M ON W (a2);
  M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
    ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
    ILWHVLQ IMWHVLQ IHWHVLQ
    ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
    ILWHVMQ IMWHVMQ IHWHVMQ
    ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
    ILWHVHQ IMWHVHQ IHWHVHQ
    DIR_LOWW DIR_MEDW DIR_HIW
    TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
    TLWHVLQ TMWHVLQ THWHVLQ
    TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
    TLWHVMQ TMWHVMQ THWHVMQ
    TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
    TLWHVHQ TMWHVHQ THWHVHQ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
```

```
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q;

    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q;

    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q;
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q;

    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q;

    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
```

```
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q;
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q;

    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q;

    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q;

    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q;
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q;
```

```
! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWW = cdash1 + cdash3*LOW_W;
    DIR_MEDW = cdash1 + cdash3*MED_W;
    DIR_HIW = cdash1 + cdash3*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TLWLVLQ = ILWLVLQ + DIR_LOWW;
    TMWLVLQ = IMWLVLQ + DIR_MEDW;
    THWLVLQ = IHWLVLQ + DIR_HIW;

    TLWMVLQ = ILWMVLQ + DIR_LOWW;
    TMWMVLQ = IMWMVLQ + DIR_MEDW;
    THWMVLQ = IHWMVLQ + DIR_HIW;

    TLWHVLQ = ILWHVLQ + DIR_LOWW;
    TMWHVLQ = IMWHVLQ + DIR_MEDW;
    THWHVLQ = IHWHVLQ + DIR_HIW;

    TLWLVMQ = ILWLVMQ + DIR_LOWW;
    TMWLVMQ = IMWLVMQ + DIR_MEDW;
    THWLVMQ = IHWLVMQ + DIR_HIW;

    TLWMVMQ = ILWMVMQ + DIR_LOWW;
    TMWMVMQ = IMWMVMQ + DIR_MEDW;
    THWMVMQ = IHWMVMQ + DIR_HIW;

    TLWHVMQ = ILWHVMQ + DIR_LOWW;
    TMWHVMQ = IMWHVMQ + DIR_MEDW;
    THWHVMQ = IHWHVMQ + DIR_HIW;

    TLWLVHQ = ILWLVHQ + DIR_LOWW;
    TMWLVHQ = IMWLVHQ + DIR_MEDW;
    THWLVHQ = IHWLVHQ + DIR_HIW;

    TLWMVHQ = ILWMVHQ + DIR_LOWW;
    TMWMVHQ = IMWMVHQ + DIR_MEDW;
    THWMVHQ = IHWMVHQ + DIR_HIW;

    TLWHVHQ = ILWHVHQ + DIR_LOWW;
    TMWHVHQ = IMWHVHQ + DIR_MEDW;
    THWHVHQ = IHWHVHQ + DIR_HIW;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis
```

```
    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
    PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
    PLWHVMQ PMWHVMQ PHWHVMQ
    PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
    PLWHVHQ PMWHVHQ PHWHVHQ);

    LOOP(XVAL,1,5,0.1);

    PLWLVLQ = ILWLVLQ*XVAL;
    PMWLVLQ = IMWLVLQ*XVAL;
    PHWLVLQ = IHWLVLQ*XVAL;

    PLWMVLQ = ILWMVLQ*XVAL;
    PMWMVLQ = IMWMVLQ*XVAL;
    PHWMVLQ = IHWMVLQ*XVAL;

    PLWHVLQ = ILWHVLQ*XVAL;
    PMWHVLQ = IMWHVLQ*XVAL;
    PHWHVLQ = IHWHVLQ*XVAL;

    PLWLVMQ = ILWLVMQ*XVAL;
    PMWLVMQ = IMWLVMQ*XVAL;
    PHWLVMQ = IHWLVMQ*XVAL;

    PLWMVMQ = ILWMVMQ*XVAL;
    PMWMVMQ = IMWMVMQ*XVAL;
    PHWMVMQ = IHWMVMQ*XVAL;

    PLWHVMQ = ILWHVMQ*XVAL;
    PMWHVMQ = IMWHVMQ*XVAL;
    PHWHVMQ = IHWHVMQ*XVAL;

    PLWLVHQ = ILWLVHQ*XVAL;
    PMWLVHQ = IMWLVHQ*XVAL;
    PHWLVHQ = IHWLVHQ*XVAL;

    PLWMVHQ = ILWMVHQ*XVAL;
    PMWMVHQ = IMWMVHQ*XVAL;
    PHWMVHQ = IHWMVHQ*XVAL;

    PLWHVHQ = ILWHVHQ*XVAL;
    PMWHVHQ = IMWHVHQ*XVAL;
    PHWHVHQ = IHWHVHQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 41: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV- Mediator path and the direct IV-DV path, 2 moderating both the Mediator-DV path and the direct IV-DV path

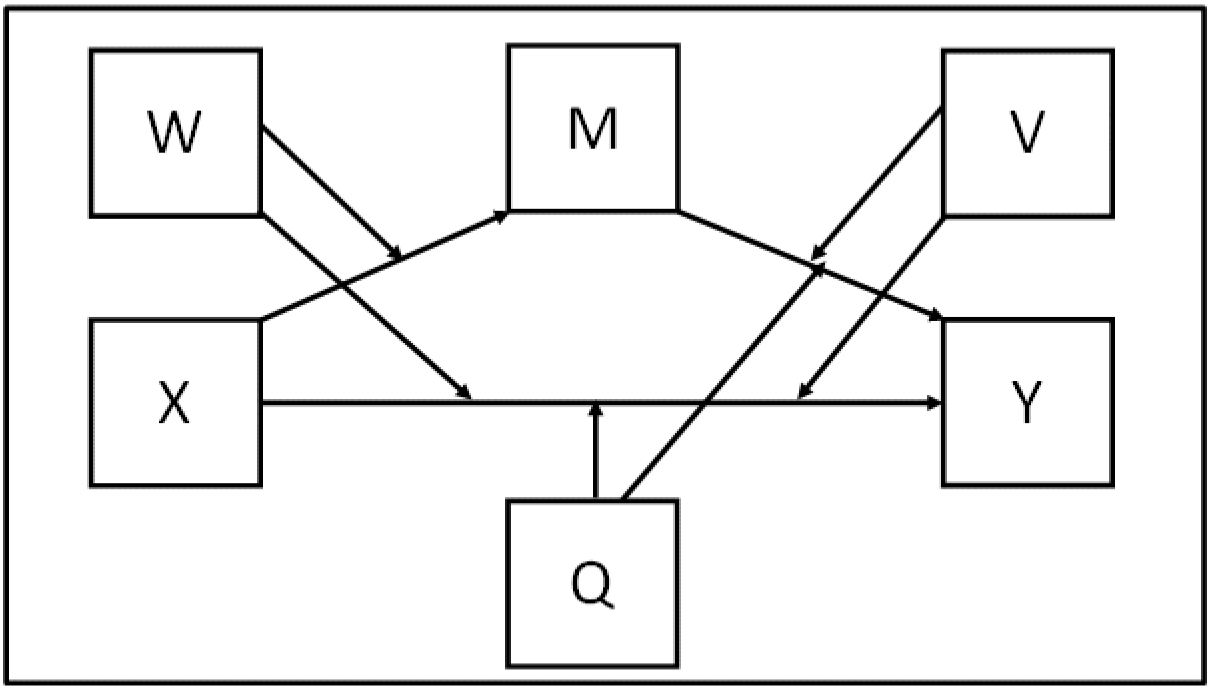
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

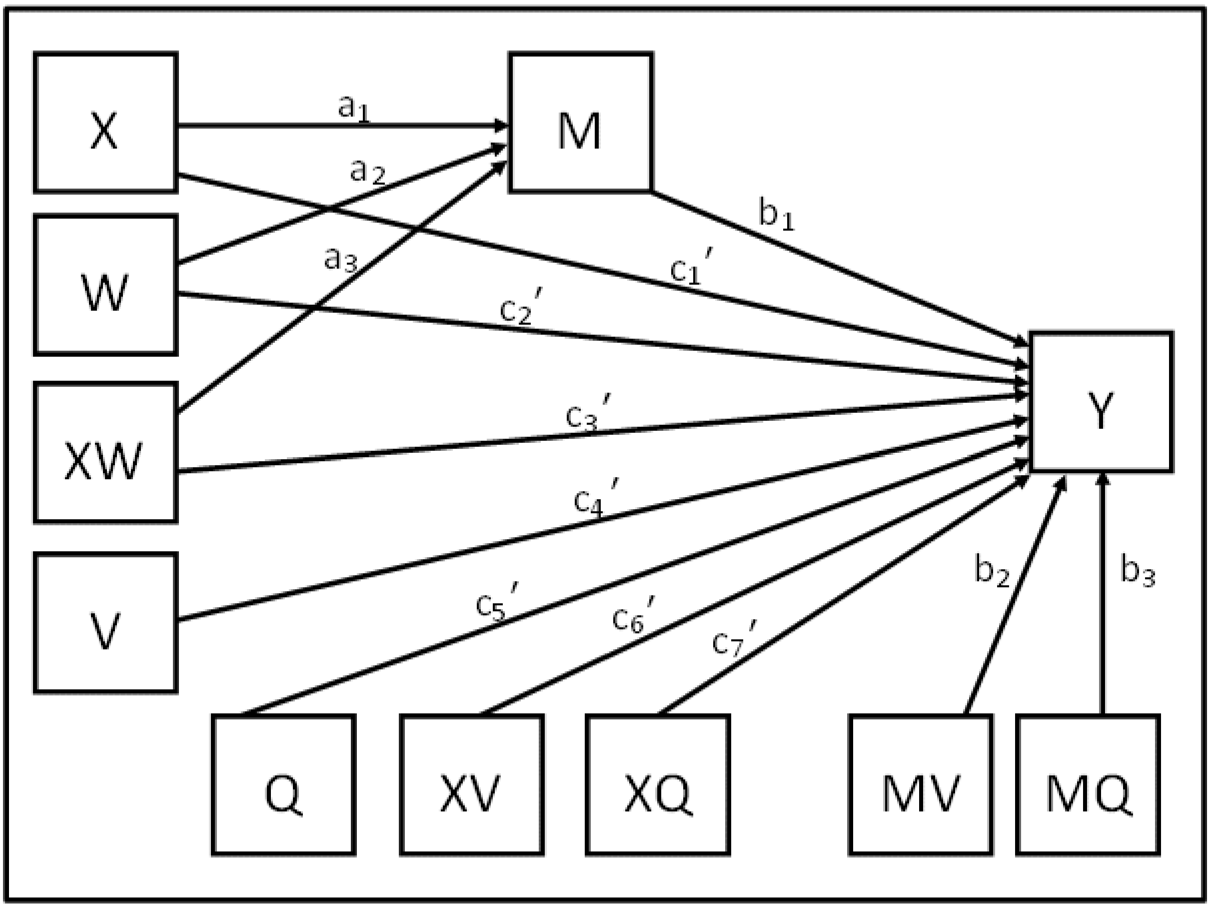
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'W + c3'XW + c4'V + c5'Q + c6'XV + c7'XQ
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'W + c3'XW + c4'V + c5'Q + c6'XV + c7'XQ
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2(a0 + a1X + a2W + a3XW)V + b3(a0 + a1X + a2W + a3XW)Q + c1'X + c2'W + c3'XW + c4'V + c5'Q + c6'XV + c7'XQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + a0b2V + a1b2XV + a2b2WV + a3b2XWV + a0b3Q + a1b3XQ + a2b3WQ + a3b3XWQ + c1'X + c2'W + c3'XW + c4'V + c5'Q + c6'XV + c7'XQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a0b2V + a2b2WV + a0b3Q + a2b3WQ + c2'W + c4'V + c5'Q) + (a1b1 + a3b1W + a1b2V + a3b2WV + a1b3Q + a3b3WQ + c1' + c3'W + c6'V + c7'Q)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V, Q:

a1b1 + a3b1W + a1b2V + a3b2WV + a1b3Q + a3b3WQ = (a1 + a3W)(b1 + b2V + b3Q)

One direct effect of X on Y, conditional on W, V, Q:

c1' + c3'W + c6'V + c7'Q

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW XV XQ MV MQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XQ = X*Q;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON MV (b2);
    Y ON MQ (b3);

    Y ON X (cdash1);
    Y ON W (cdash2);
    Y ON XW (cdash3);
    Y ON V (cdash4);
    Y ON Q (cdash5);
    Y ON XV (cdash6);
    Y ON XQ (cdash7);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);
! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
     NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
     ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
     ILWHVLQ IMWHVLQ IHWHVLQ
     ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
     ILWHVMQ IMWHVMQ IHWHVMQ
     ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
     ILWHVHQ IMWHVHQ IHWHVHQ
     DLWLVLQ DMWLVLQ DHWLVLQ DLWMVLQ DMWMVLQ DHWMVLQ
     DLWHVLQ DMWHVLQ DHWHVLQ
     DLWLVMQ DMWLVMQ DHWLVMQ DLWMVMQ DMWMVMQ DHWMVMQ
     DLWHVMQ DMWHVMQ DHWHVMQ
     DLWLVHQ DMWLVHQ DHWLVHQ DLWMVHQ DMWMVHQ DHWMVHQ
     DLWHVHQ DMWHVHQ DHWHVHQ
     TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
     TLWHVLQ TMWHVLQ THWHVLQ
     TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
```

```
    TLWHVMQ TMWHVMQ THWHVMQ
    TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
    TLWHVHQ TMWHVHQ THWHVHQ);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q;

    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q;

    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
```

```
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q;
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q;

    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q;

    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q;
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q;

    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q;
```

```
    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q;

    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q;
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLWLVLQ = cdash1 + cdash3*LOW_W + cdash6*LOW_V +
cdash7*LOW_Q;
    DMWLVLQ = cdash1 + cdash3*MED_W + cdash6*LOW_V +
cdash7*LOW_Q;
    DHWLVLQ = cdash1 + cdash3*HIGH_W + cdash6*LOW_V +
cdash7*LOW_Q;

    DLWMVLQ = cdash1 + cdash3*LOW_W + cdash6*MED_V +
cdash7*LOW_Q;
    DMWMVLQ = cdash1 + cdash3*MED_W + cdash6*MED_V +
cdash7*LOW_Q;
    DHWMVLQ = cdash1 + cdash3*HIGH_W + cdash6*MED_V +
cdash7*LOW_Q;

    DLWHVLQ = cdash1 + cdash3*LOW_W + cdash6*HIGH_V +
cdash7*LOW_Q;
    DMWHVLQ = cdash1 + cdash3*MED_W + cdash6*HIGH_V +
cdash7*LOW_Q;
    DHWHVLQ = cdash1 + cdash3*HIGH_W + cdash6*HIGH_V +
cdash7*LOW_Q;

    DLWLVMQ = cdash1 + cdash3*LOW_W + cdash6*LOW_V +
cdash7*MED_Q;
    DMWLVMQ = cdash1 + cdash3*MED_W + cdash6*LOW_V +
cdash7*MED_Q;
    DHWLVMQ = cdash1 + cdash3*HIGH_W + cdash6*LOW_V +
cdash7*MED_Q;
```

```
    DLWMVMQ = cdash1 + cdash3*LOW_W + cdash6*MED_V +
cdash7*MED_Q;
    DMWMVMQ = cdash1 + cdash3*MED_W + cdash6*MED_V +
cdash7*MED_Q;
    DHWMVMQ = cdash1 + cdash3*HIGH_W + cdash6*MED_V +
cdash7*MED_Q;

    DLWHVMQ = cdash1 + cdash3*LOW_W + cdash6*HIGH_V +
cdash7*MED_Q;
    DMWHVMQ = cdash1 + cdash3*MED_W + cdash6*HIGH_V +
cdash7*MED_Q;
    DHWHVMQ = cdash1 + cdash3*HIGH_W + cdash6*HIGH_V +
cdash7*MED_Q;

    DLWLVHQ = cdash1 + cdash3*LOW_W + cdash6*LOW_V +
cdash7*HIGH_Q;
    DMWLVHQ = cdash1 + cdash3*MED_W + cdash6*LOW_V +
cdash7*HIGH_Q;
    DHWLVHQ = cdash1 + cdash3*HIGH_W + cdash6*LOW_V +
cdash7*HIGH_Q;

    DLWMVHQ = cdash1 + cdash3*LOW_W + cdash6*MED_V +
cdash7*HIGH_Q;
    DMWMVHQ = cdash1 + cdash3*MED_W + cdash6*MED_V +
cdash7*HIGH_Q;
    DHWMVHQ = cdash1 + cdash3*HIGH_W + cdash6*MED_V +
cdash7*HIGH_Q;

    DLWHVHQ = cdash1 + cdash3*LOW_W + cdash6*HIGH_V +
cdash7*HIGH_Q;
    DMWHVHQ = cdash1 + cdash3*MED_W + cdash6*HIGH_V +
cdash7*HIGH_Q;
    DHWHVHQ = cdash1 + cdash3*HIGH_W + cdash6*HIGH_V +
cdash7*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLWLVLQ = ILWLVLQ + DLWLVLQ;
    TMWLVLQ = IMWLVLQ + DMWLVLQ;
    THWLVLQ = IHWLVLQ + DHWLVLQ;

    TLWMVLQ = ILWMVLQ + DLWMVLQ;
    TMWMVLQ = IMWMVLQ + DMWMVLQ;
    THWMVLQ = IHWMVLQ + DHWMVLQ;

    TLWHVLQ = ILWHVLQ + DLWHVLQ;
    TMWHVLQ = IMWHVLQ + DMWHVLQ;
    THWHVLQ = IHWHVLQ + DHWHVLQ;
```

```
    TLWLVMQ = ILWLVMQ + DLWLVMQ;
    TMWLVMQ = IMWLVMQ + DMWLVMQ;
    THWLVMQ = IHWLVMQ + DHWLVMQ;

    TLWMVMQ = ILWMVMQ + DLWMVMQ;
    TMWMVMQ = IMWMVMQ + DMWMVMQ;
    THWMVMQ = IHWMVMQ + DHWMVMQ;

    TLWHVMQ = ILWHVMQ + DLWHVMQ;
    TMWHVMQ = IMWHVMQ + DMWHVMQ;
    THWHVMQ = IHWHVMQ + DHWHVMQ;

    TLWLVHQ = ILWLVHQ + DLWLVHQ;
    TMWLVHQ = IMWLVHQ + DMWLVHQ;
    THWLVHQ = IHWLVHQ + DHWLVHQ;

    TLWMVHQ = ILWMVHQ + DLWMVHQ;
    TMWMVHQ = IMWMVHQ + DMWMVHQ;
    THWMVHQ = IHWMVHQ + DHWMVHQ;

    TLWHVHQ = ILWHVHQ + DLWHVHQ;
    TMWHVHQ = IMWHVHQ + DMWHVHQ;
    THWHVHQ = IHWHVHQ + DHWHVHQ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
    PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
    PLWHVMQ PMWHVMQ PHWHVMQ
    PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
    PLWHVHQ PMWHVHQ PHWHVHQ);

    LOOP(XVAL,1,5,0.1);

    PLWLVLQ = ILWLVLQ*XVAL;
    PMWLVLQ = IMWLVLQ*XVAL;
    PHWLVLQ = IHWLVLQ*XVAL;

    PLWMVLQ = ILWMVLQ*XVAL;
    PMWMVLQ = IMWMVLQ*XVAL;
    PHWMVLQ = IHWMVLQ*XVAL;

    PLWHVLQ = ILWHVLQ*XVAL;
    PMWHVLQ = IMWHVLQ*XVAL;
    PHWHVLQ = IHWHVLQ*XVAL;
```

```
    PLWLVMQ = ILWLVMQ*XVAL;
    PMWLVMQ = IMWLVMQ*XVAL;
    PHWLVMQ = IHWLVMQ*XVAL;

    PLWMVMQ = ILWMVMQ*XVAL;
    PMWMVMQ = IMWMVMQ*XVAL;
    PHWMVMQ = IHWMVMQ*XVAL;

    PLWHVMQ = ILWHVMQ*XVAL;
    PMWHVMQ = IMWHVMQ*XVAL;
    PHWHVMQ = IHWHVMQ*XVAL;

    PLWLVHQ = ILWLVHQ*XVAL;
    PMWLVHQ = IMWLVHQ*XVAL;
    PHWLVHQ = IHWLVHQ*XVAL;

    PLWMVHQ = ILWMVHQ*XVAL;
    PMWMVHQ = IMWMVHQ*XVAL;
    PHWMVHQ = IHWMVHQ*XVAL;

    PLWHVHQ = ILWHVHQ*XVAL;
    PMWHVHQ = IMWHVHQ*XVAL;
    PHWHVHQ = IHWHVHQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 42: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV- Mediator path and the direct IV-DV path, 2 moderating the Mediator-DV path all 2-way and 3-way interactions

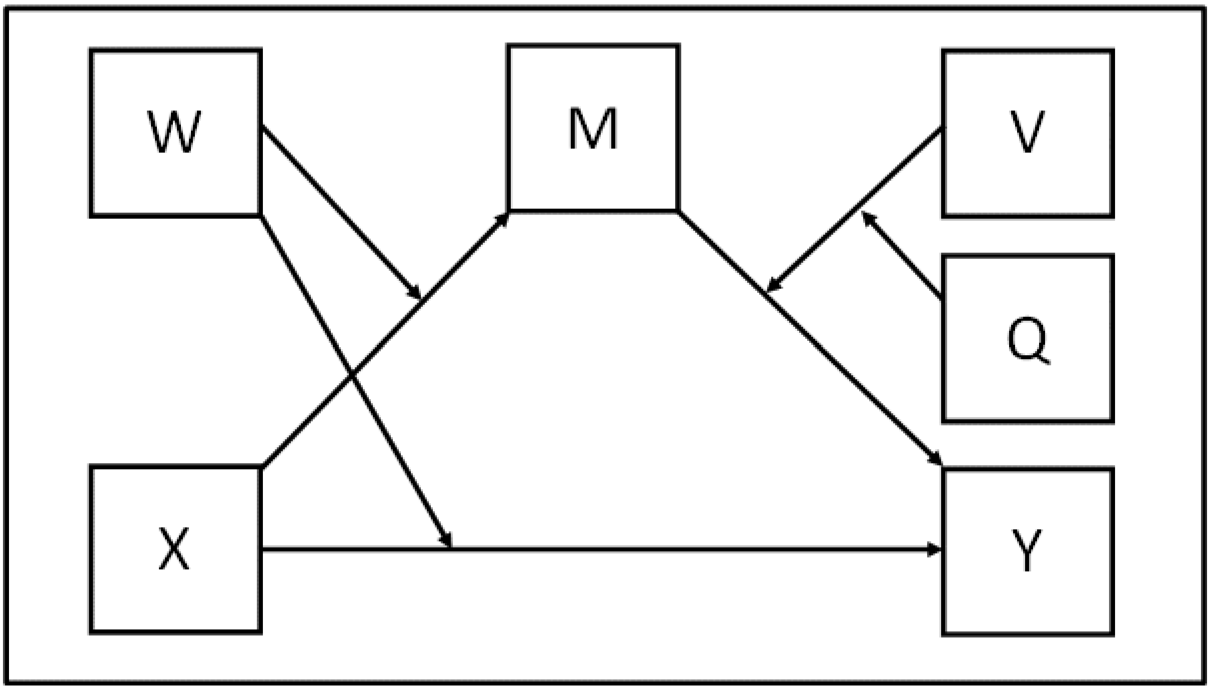
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

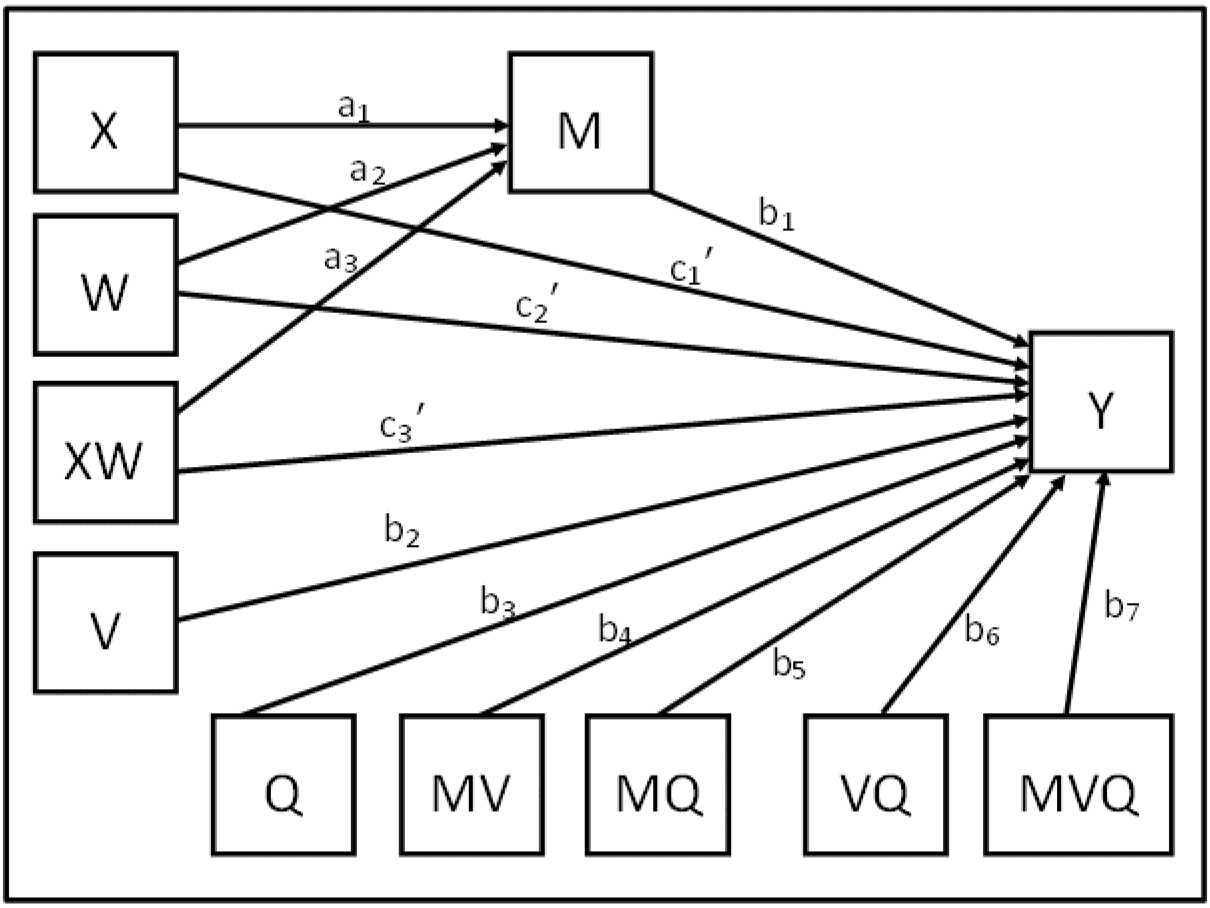
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2V + b3Q + b4(a0 + a1X + a2W + a3XW)V + b5(a0 + a1X + a2W + a3XW)Q + b6VQ + b7(a0 + a1X + a2W + a3XW)VQ + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4XWV + a0b5 + a1b5XQ + a2b5WQ + a3b5XWQ + b6VQ + a0b7VQ + a1b7XVQ + a2b7WVQ + a3b7XWVQ + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2V + b3Q + a0b4V + a2b4WV + a0b5 + a2b5WQ + b6VQ + a0b7VQ + a2b7WVQ + c2'W) + (a1b1 + a3b1W + a1b4V + a3b4WV + a1b5Q + a3b5WQ + a1b7VQ + a3b7WVQ + c1' + c3'W)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V, Q:

a1b1 + a3b1W + a1b4V + a3b4WV + a1b5Q + a3b5WQ + a1b7VQ + a3b7WVQ = (a1 + a3W)(b1 + b4V + b5Q + b7VQ)

One direct effect of X on Y, conditional on W:

c1' + c3'W

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW VQ MV MQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   VQ = V*Q;
   MVQ = M*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;
```

```
! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);
   Y ON VQ (b6);
   Y ON MVQ (b7);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
    ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
    ILWHVLQ IMWHVLQ IHWHVLQ
    ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
    ILWHVMQ IMWHVMQ IHWHVMQ
    ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
    ILWHVHQ IMWHVHQ IHWHVHQ
    DIR_LOWW DIR_MEDW DIR_HIW
    TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
    TLWHVLQ TMWHVLQ THWHVLQ
    TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
    TLWHVMQ TMWHVMQ THWHVMQ
    TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
    TLWHVHQ TMWHVHQ THWHVHQ);
```

```
    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q + a1*b7*LOW_V*LOW_Q +
     a3*b7*LOW_W*LOW_V*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q + a1*b7*LOW_V*LOW_Q +
     a3*b7*MED_W*LOW_V*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q + a1*b7*LOW_V*LOW_Q +
     a3*b7*HIGH_W*LOW_V*LOW_Q;

    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q + a1*b7*MED_V*LOW_Q +
     a3*b7*LOW_W*MED_V*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q + a1*b7*MED_V*LOW_Q +
     a3*b7*MED_W*MED_V*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q + a1*b7*MED_V*LOW_Q +
     a3*b7*HIGH_W*MED_V*LOW_Q;
```

```
    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*LOW_W*LOW_Q + a1*b7*HIGH_V*LOW_Q +
     a3*b7*LOW_W*HIGH_V*LOW_Q;
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*MED_W*LOW_Q + a1*b7*HIGH_V*LOW_Q +
     a3*b7*MED_W*HIGH_V*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*LOW_Q + a3*b5*HIGH_W*LOW_Q + a1*b7*HIGH_V*LOW_Q +
     a3*b7*HIGH_W*HIGH_V*LOW_Q;

    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q + a1*b7*LOW_V*MED_Q +
     a3*b7*LOW_W*LOW_V*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q + a1*b7*LOW_V*MED_Q +
     a3*b7*MED_W*LOW_V*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q + a1*b7*LOW_V*MED_Q +
     a3*b7*HIGH_W*LOW_V*MED_Q;

    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q + a1*b7*MED_V*MED_Q +
     a3*b7*LOW_W*MED_V*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q + a1*b7*MED_V*MED_Q +
     a3*b7*MED_W*MED_V*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q + a1*b7*MED_V*MED_Q +
     a3*b7*HIGH_W*MED_V*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*LOW_W*MED_Q + a1*b7*HIGH_V*MED_Q +
     a3*b7*LOW_W*HIGH_V*MED_Q;
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*MED_Q + a3*b5*MED_W*MED_Q + a1*b7*HIGH_V*MED_Q +
     a3*b7*MED_W*HIGH_V*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
```

```
     a1*b5*MED_Q + a3*b5*HIGH_W*MED_Q + a1*b7*HIGH_V*MED_Q +
     a3*b7*HIGH_W*HIGH_V*MED_Q;

    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_V +
a3*b4*LOW_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q + a1*b7*LOW_V*HIGH_Q +
     a3*b7*LOW_W*LOW_V*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b4*LOW_V +
a3*b4*MED_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q + a1*b7*LOW_V*HIGH_Q +
     a3*b7*MED_W*LOW_V*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*LOW_V +
a3*b4*HIGH_W*LOW_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q + a1*b7*LOW_V*HIGH_Q
+
     a3*b7*HIGH_W*LOW_V*HIGH_Q;

    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*MED_V +
a3*b4*LOW_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q + a1*b7*MED_V*HIGH_Q +
     a3*b7*LOW_W*MED_V*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b4*MED_V +
a3*b4*MED_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q + a1*b7*MED_V*HIGH_Q +
     a3*b7*MED_W*MED_V*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*MED_V +
a3*b4*HIGH_W*MED_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q + a1*b7*MED_V*HIGH_Q
+
     a3*b7*HIGH_W*MED_V*HIGH_Q;

    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b4*HIGH_V +
a3*b4*LOW_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*LOW_W*HIGH_Q + a1*b7*HIGH_V*HIGH_Q
+
     a3*b7*LOW_W*HIGH_V*HIGH_Q;
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b4*HIGH_V +
a3*b4*MED_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*MED_W*HIGH_Q + a1*b7*HIGH_V*HIGH_Q
+
     a3*b7*MED_W*HIGH_V*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_V +
a3*b4*HIGH_W*HIGH_V +
     a1*b5*HIGH_Q + a3*b5*HIGH_W*HIGH_Q + a1*b7*HIGH_V*HIGH_Q
+
     a3*b7*HIGH_W*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values
```

```
    DIR_LOWW = cdash1 + cdash3*LOW_W;
    DIR_MEDW = cdash1 + cdash3*MED_W;
    DIR_HIW = cdash1 + cdash3*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TLWLVLQ = ILWLVLQ + DIR_LOWW;
    TMWLVLQ = IMWLVLQ + DIR_MEDW;
    THWLVLQ = IHWLVLQ + DIR_HIW;

    TLWMVLQ = ILWMVLQ + DIR_LOWW;
    TMWMVLQ = IMWMVLQ + DIR_MEDW;
    THWMVLQ = IHWMVLQ + DIR_HIW;

    TLWHVLQ = ILWHVLQ + DIR_LOWW;
    TMWHVLQ = IMWHVLQ + DIR_MEDW;
    THWHVLQ = IHWHVLQ + DIR_HIW;

    TLWLVMQ = ILWLVMQ + DIR_LOWW;
    TMWLVMQ = IMWLVMQ + DIR_MEDW;
    THWLVMQ = IHWLVMQ + DIR_HIW;

    TLWMVMQ = ILWMVMQ + DIR_LOWW;
    TMWMVMQ = IMWMVMQ + DIR_MEDW;
    THWMVMQ = IHWMVMQ + DIR_HIW;

    TLWHVMQ = ILWHVMQ + DIR_LOWW;
    TMWHVMQ = IMWHVMQ + DIR_MEDW;
    THWHVMQ = IHWHVMQ + DIR_HIW;

    TLWLVHQ = ILWLVHQ + DIR_LOWW;
    TMWLVHQ = IMWLVHQ + DIR_MEDW;
    THWLVHQ = IHWLVHQ + DIR_HIW;

    TLWMVHQ = ILWMVHQ + DIR_LOWW;
    TMWMVHQ = IMWMVHQ + DIR_MEDW;
    THWMVHQ = IHWMVHQ + DIR_HIW;

    TLWHVHQ = ILWHVHQ + DIR_LOWW;
    TMWHVHQ = IMWHVHQ + DIR_MEDW;
    THWHVHQ = IHWHVHQ + DIR_HIW;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
```

```
     PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
     PLWHVMQ PMWHVMQ PHWHVMQ
     PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
     PLWHVHQ PMWHVHQ PHWHVHQ);

     LOOP(XVAL,1,5,0.1);

     PLWLVLQ = ILWLVLQ*XVAL;
     PMWLVLQ = IMWLVLQ*XVAL;
     PHWLVLQ = IHWLVLQ*XVAL;

     PLWMVLQ = ILWMVLQ*XVAL;
     PMWMVLQ = IMWMVLQ*XVAL;
     PHWMVLQ = IHWMVLQ*XVAL;

     PLWHVLQ = ILWHVLQ*XVAL;
     PMWHVLQ = IMWHVLQ*XVAL;
     PHWHVLQ = IHWHVLQ*XVAL;

     PLWLVMQ = ILWLVMQ*XVAL;
     PMWLVMQ = IMWLVMQ*XVAL;
     PHWLVMQ = IHWLVMQ*XVAL;

     PLWMVMQ = ILWMVMQ*XVAL;
     PMWMVMQ = IMWMVMQ*XVAL;
     PHWMVMQ = IHWMVMQ*XVAL;

     PLWHVMQ = ILWHVMQ*XVAL;
     PMWHVMQ = IMWHVMQ*XVAL;
     PHWHVMQ = IHWHVMQ*XVAL;

     PLWLVHQ = ILWLVHQ*XVAL;
     PMWLVHQ = IMWLVHQ*XVAL;
     PHWLVHQ = IHWLVHQ*XVAL;

     PLWMVHQ = ILWMVHQ*XVAL;
     PMWMVHQ = IMWMVHQ*XVAL;
     PHWMVHQ = IHWMVHQ*XVAL;

     PLWHVHQ = ILWHVHQ*XVAL;
     PMWHVHQ = IMWHVHQ*XVAL;
     PHWHVHQ = IHWHVHQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 43: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV- Mediator path and the direct IV-DV path, 2 moderating both the Mediator-DV path and the direct IV-DV path with all 2-way and 3-way interactions

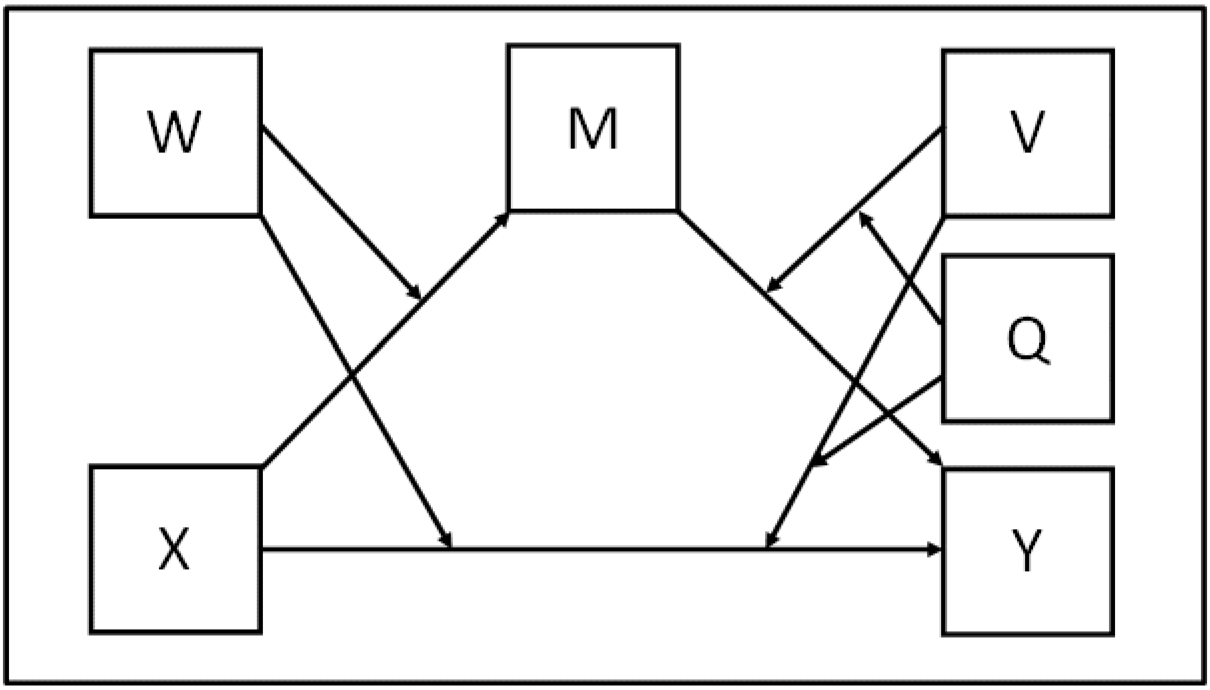
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

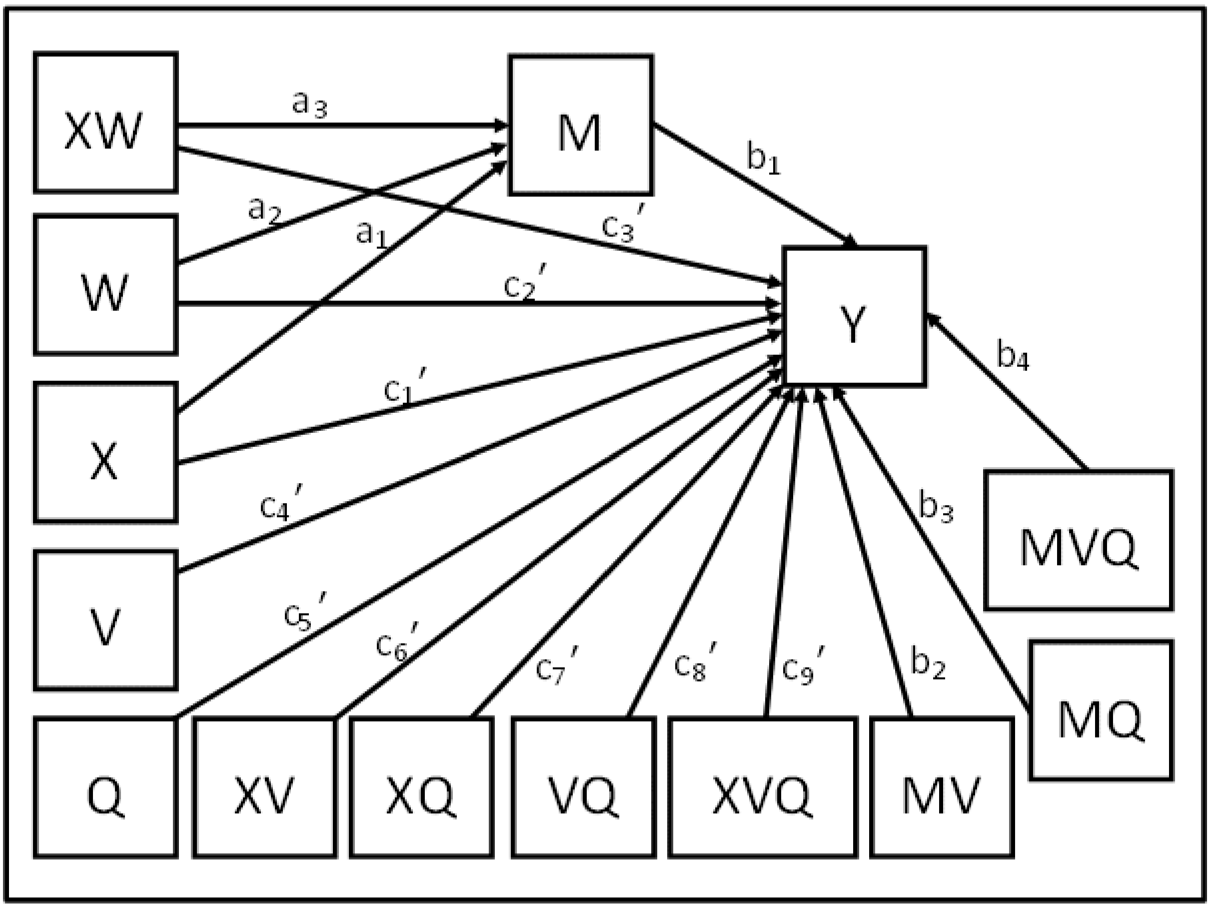
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'W + c3'XW + c4'V + c5'Q + c6'XV + c7'XQ + c8'VQ + c9'XVQ
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'W + c3'XW + c4'V + c5'Q + c6'XV + c7'XQ + c8'VQ + c9'XVQ
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2(a0 + a1X + a2W + a3XW)V + b3(a0 + a1X + a2W + a3XW)Q + b4(a0 + a1X + a2W + a3XW)VQ + c1'X + c2'W + c3'XW + c4'V + c5'Q + c6'XV + c7'XQ + c8'VQ + c9'XVQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + a0b2V + a1b2XV + a2b2WV + a3b2XWV + a0b3Q + a1b3XQ + a2b3WQ + a3b3XWQ + a0b4VQ + a1b4XVQ + a2b4WVQ + a3b4XWVQ + c1'X + c2'W + c3'XW + c4'V + c5'Q + c6'XV + c7'XQ + c8'VQ + c9'XVQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a0b2V + a2b2WV + a0b3Q + a2b3WQ + a0b4VQ + a2b4WVQ + c2'W + c4'V + c5'Q + c8'VQ) + (a1b1 + a3b1W + a1b2V + a3b2WV + a1b3Q + a3b3WQ + a1b4VQ + a3b4WVQ + c1' + c3'W + c6'V + c7'Q + c9'VQ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V, Q:

a1b1 + a3b1W + a1b2V + a3b2WV + a1b3Q + a3b3WQ + a1b4VQ + a3b4WVQ = (a1 + a3W)(b1 + b2V + b3Q + b4VQ)

One direct effect of X on Y, conditional on W, V, Q:

c1' + c3'W + c6'V + c7'Q + c9'VQ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW XV XQ VQ MV MQ XVQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XQ = X*Q;
   XV = X*V;
   VQ = V*Q;
   MVQ = M*V*Q;
   XVQ = X*V*Q;

ANALYSIS:
   TYPE = GENERAL;
```

```
    ESTIMATOR = ML;
    BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON MV (b2);
    Y ON MQ (b3);
    Y ON MVQ (b4);

    Y ON X (cdash1);
    Y ON W (cdash2);
    Y ON XW (cdash3);
    Y ON V (cdash4);
    Y ON Q (cdash5);
    Y ON XV (cdash6);
    Y ON XQ (cdash7);
    Y ON VQ (cdash8);
    Y ON XVQ (cdash9);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
    ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
    ILWHVLQ IMWHVLQ IHWHVLQ
    ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
    ILWHVMQ IMWHVMQ IHWHVMQ
    ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
    ILWHVHQ IMWHVHQ IHWHVHQ
    DLWLVLQ DMWLVLQ DHWLVLQ DLWMVLQ DMWMVLQ DHWMVLQ
```

```
    DLWHVLQ DMWHVLQ DHWHVLQ
    DLWLVMQ DMWLVMQ DHWLVMQ DLWMVMQ DMWMVMQ DHWMVMQ
    DLWHVMQ DMWHVMQ DHWHVMQ
    DLWLVHQ DMWLVHQ DHWLVHQ DLWMVHQ DMWMVHQ DHWMVHQ
    DLWHVHQ DMWHVHQ DHWHVHQ
    TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
    TLWHVLQ TMWHVLQ THWHVLQ
    TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
    TLWHVMQ TMWHVMQ THWHVMQ
    TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
    TLWHVHQ TMWHVHQ THWHVHQ);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q + a1*b4*LOW_V*LOW_Q +
     a3*b4*LOW_W*LOW_V*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q + a1*b4*LOW_V*LOW_Q +
     a3*b4*MED_W*LOW_V*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q + a1*b4*LOW_V*LOW_Q +
     a3*b4*HIGH_W*LOW_V*LOW_Q;
```

```
    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q + a1*b4*MED_V*LOW_Q +
     a3*b4*LOW_W*MED_V*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q + a1*b4*MED_V*LOW_Q +
     a3*b4*MED_W*MED_V*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q + a1*b4*MED_V*LOW_Q +
     a3*b4*HIGH_W*MED_V*LOW_Q;

    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*LOW_W*LOW_Q + a1*b4*HIGH_V*LOW_Q +
     a3*b4*LOW_W*HIGH_V*LOW_Q;
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*MED_W*LOW_Q + a1*b4*HIGH_V*LOW_Q +
     a3*b4*MED_W*HIGH_V*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*LOW_Q + a3*b3*HIGH_W*LOW_Q + a1*b4*HIGH_V*LOW_Q +
     a3*b4*HIGH_W*HIGH_V*LOW_Q;

    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q + a1*b4*LOW_V*MED_Q +
     a3*b4*LOW_W*LOW_V*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q + a1*b4*LOW_V*MED_Q +
     a3*b4*MED_W*LOW_V*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q + a1*b4*LOW_V*MED_Q +
     a3*b4*HIGH_W*LOW_V*MED_Q;

    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q + a1*b4*MED_V*MED_Q +
     a3*b4*LOW_W*MED_V*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q + a1*b4*MED_V*MED_Q +
     a3*b4*MED_W*MED_V*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
```

```
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q + a1*b4*MED_V*MED_Q +
     a3*b4*HIGH_W*MED_V*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*LOW_W*MED_Q + a1*b4*HIGH_V*MED_Q +
     a3*b4*LOW_W*HIGH_V*MED_Q;
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*MED_W*MED_Q + a1*b4*HIGH_V*MED_Q +
     a3*b4*MED_W*HIGH_V*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*MED_Q + a3*b3*HIGH_W*MED_Q + a1*b4*HIGH_V*MED_Q +
     a3*b4*HIGH_W*HIGH_V*MED_Q;

    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_V +
a3*b2*LOW_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q + a1*b4*LOW_V*HIGH_Q +
     a3*b4*LOW_W*LOW_V*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b2*LOW_V +
a3*b2*MED_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q + a1*b4*LOW_V*HIGH_Q +
     a3*b4*MED_W*LOW_V*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*LOW_V +
a3*b2*HIGH_W*LOW_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q + a1*b4*LOW_V*HIGH_Q
+
     a3*b4*HIGH_W*LOW_V*HIGH_Q;

    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*MED_V +
a3*b2*LOW_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q + a1*b4*MED_V*HIGH_Q +
     a3*b4*LOW_W*MED_V*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b2*MED_V +
a3*b2*MED_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q + a1*b4*MED_V*HIGH_Q +
     a3*b4*MED_W*MED_V*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*MED_V +
a3*b2*HIGH_W*MED_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q + a1*b4*MED_V*HIGH_Q
+
     a3*b4*HIGH_W*MED_V*HIGH_Q;

    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b2*HIGH_V +
a3*b2*LOW_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*LOW_W*HIGH_Q + a1*b4*HIGH_V*HIGH_Q
+
     a3*b4*LOW_W*HIGH_V*HIGH_Q;
```

```
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b2*HIGH_V +
a3*b2*MED_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*MED_W*HIGH_Q + a1*b4*HIGH_V*HIGH_Q
+
     a3*b4*MED_W*HIGH_V*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_V +
a3*b2*HIGH_W*HIGH_V +
     a1*b3*HIGH_Q + a3*b3*HIGH_W*HIGH_Q + a1*b4*HIGH_V*HIGH_Q
+
     a3*b4*HIGH_W*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLWLVLQ = cdash1 + cdash3*LOW_W + cdash6*LOW_V +
cdash7*LOW_Q +
     cdash9*LOW_V*LOW_Q;
    DMWLVLQ = cdash1 + cdash3*MED_W + cdash6*LOW_V +
cdash7*LOW_Q +
     cdash9*LOW_V*LOW_Q;
    DHWLVLQ = cdash1 + cdash3*HIGH_W + cdash6*LOW_V +
cdash7*LOW_Q +
     cdash9*LOW_V*LOW_Q;

    DLWMVLQ = cdash1 + cdash3*LOW_W + cdash6*MED_V +
cdash7*LOW_Q +
     cdash9*MED_V*LOW_Q;
    DMWMVLQ = cdash1 + cdash3*MED_W + cdash6*MED_V +
cdash7*LOW_Q +
     cdash9*MED_V*LOW_Q;
    DHWMVLQ = cdash1 + cdash3*HIGH_W + cdash6*MED_V +
cdash7*LOW_Q +
     cdash9*MED_V*LOW_Q;

    DLWHVLQ = cdash1 + cdash3*LOW_W + cdash6*HIGH_V +
cdash7*LOW_Q +
     cdash9*HIGH_V*LOW_Q;
    DMWHVLQ = cdash1 + cdash3*MED_W + cdash6*HIGH_V +
cdash7*LOW_Q +
     cdash9*HIGH_V*LOW_Q;
    DHWHVLQ = cdash1 + cdash3*HIGH_W + cdash6*HIGH_V +
cdash7*LOW_Q +
     cdash9*HIGH_V*LOW_Q;

    DLWLVMQ = cdash1 + cdash3*LOW_W + cdash6*LOW_V +
cdash7*MED_Q +
     cdash9*LOW_V*MED_Q;
    DMWLVMQ = cdash1 + cdash3*MED_W + cdash6*LOW_V +
cdash7*MED_Q +
```

```
     cdash9*LOW_V*MED_Q;
    DHWLVMQ = cdash1 + cdash3*HIGH_W + cdash6*LOW_V +
cdash7*MED_Q +
     cdash9*LOW_V*MED_Q;

    DLWMVMQ = cdash1 + cdash3*LOW_W + cdash6*MED_V +
cdash7*MED_Q +
     cdash9*MED_V*MED_Q;
    DMWMVMQ = cdash1 + cdash3*MED_W + cdash6*MED_V +
cdash7*MED_Q +
     cdash9*MED_V*MED_Q;
    DHWMVMQ = cdash1 + cdash3*HIGH_W + cdash6*MED_V +
cdash7*MED_Q +
     cdash9*MED_V*MED_Q;

    DLWHVMQ = cdash1 + cdash3*LOW_W + cdash6*HIGH_V +
cdash7*MED_Q +
     cdash9*HIGH_V*MED_Q;
    DMWHVMQ = cdash1 + cdash3*MED_W + cdash6*HIGH_V +
cdash7*MED_Q +
     cdash9*HIGH_V*MED_Q;
    DHWHVMQ = cdash1 + cdash3*HIGH_W + cdash6*HIGH_V +
cdash7*MED_Q +
     cdash9*HIGH_V*MED_Q;

    DLWLVHQ = cdash1 + cdash3*LOW_W + cdash6*LOW_V +
cdash7*HIGH_Q +
     cdash9*LOW_V*HIGH_Q;
    DMWLVHQ = cdash1 + cdash3*MED_W + cdash6*LOW_V +
cdash7*HIGH_Q +
     cdash9*LOW_V*HIGH_Q;
    DHWLVHQ = cdash1 + cdash3*HIGH_W + cdash6*LOW_V +
cdash7*HIGH_Q +
     cdash9*LOW_V*HIGH_Q;

    DLWMVHQ = cdash1 + cdash3*LOW_W + cdash6*MED_V +
cdash7*HIGH_Q +
     cdash9*MED_V*HIGH_Q;
    DMWMVHQ = cdash1 + cdash3*MED_W + cdash6*MED_V +
cdash7*HIGH_Q +
     cdash9*MED_V*HIGH_Q;
    DHWMVHQ = cdash1 + cdash3*HIGH_W + cdash6*MED_V +
cdash7*HIGH_Q +
     cdash9*MED_V*HIGH_Q;

    DLWHVHQ = cdash1 + cdash3*LOW_W + cdash6*HIGH_V +
cdash7*HIGH_Q +
     cdash9*HIGH_V*HIGH_Q;
    DMWHVHQ = cdash1 + cdash3*MED_W + cdash6*HIGH_V +
```

```
cdash7*HIGH_Q +
     cdash9*HIGH_V*HIGH_Q;
    DHWHVHQ = cdash1 + cdash3*HIGH_W + cdash6*HIGH_V +
cdash7*HIGH_Q +
     cdash9*HIGH_V*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLWLVLQ = ILWLVLQ + DLWLVLQ;
    TMWLVLQ = IMWLVLQ + DMWLVLQ;
    THWLVLQ = IHWLVLQ + DHWLVLQ;

    TLWMVLQ = ILWMVLQ + DLWMVLQ;
    TMWMVLQ = IMWMVLQ + DMWMVLQ;
    THWMVLQ = IHWMVLQ + DHWMVLQ;

    TLWHVLQ = ILWHVLQ + DLWHVLQ;
    TMWHVLQ = IMWHVLQ + DMWHVLQ;
    THWHVLQ = IHWHVLQ + DHWHVLQ;

    TLWLVMQ = ILWLVMQ + DLWLVMQ;
    TMWLVMQ = IMWLVMQ + DMWLVMQ;
    THWLVMQ = IHWLVMQ + DHWLVMQ;

    TLWMVMQ = ILWMVMQ + DLWMVMQ;
    TMWMVMQ = IMWMVMQ + DMWMVMQ;
    THWMVMQ = IHWMVMQ + DHWMVMQ;

    TLWHVMQ = ILWHVMQ + DLWHVMQ;
    TMWHVMQ = IMWHVMQ + DMWHVMQ;
    THWHVMQ = IHWHVMQ + DHWHVMQ;

    TLWLVHQ = ILWLVHQ + DLWLVHQ;
    TMWLVHQ = IMWLVHQ + DMWLVHQ;
    THWLVHQ = IHWLVHQ + DHWLVHQ;

    TLWMVHQ = ILWMVHQ + DLWMVHQ;
    TMWMVHQ = IMWMVHQ + DMWMVHQ;
    THWMVHQ = IHWMVHQ + DHWMVHQ;

    TLWHVHQ = ILWHVHQ + DLWHVHQ;
    TMWHVHQ = IMWHVHQ + DMWHVHQ;
    THWHVHQ = IHWHVHQ + DHWHVHQ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis
```

```
    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
    PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
    PLWHVMQ PMWHVMQ PHWHVMQ
    PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
    PLWHVHQ PMWHVHQ PHWHVHQ);

    LOOP(XVAL,1,5,0.1);

    PLWLVLQ = ILWLVLQ*XVAL;
    PMWLVLQ = IMWLVLQ*XVAL;
    PHWLVLQ = IHWLVLQ*XVAL;

    PLWMVLQ = ILWMVLQ*XVAL;
    PMWMVLQ = IMWMVLQ*XVAL;
    PHWMVLQ = IHWMVLQ*XVAL;

    PLWHVLQ = ILWHVLQ*XVAL;
    PMWHVLQ = IMWHVLQ*XVAL;
    PHWHVLQ = IHWHVLQ*XVAL;

    PLWLVMQ = ILWLVMQ*XVAL;
    PMWLVMQ = IMWLVMQ*XVAL;
    PHWLVMQ = IHWLVMQ*XVAL;

    PLWMVMQ = ILWMVMQ*XVAL;
    PMWMVMQ = IMWMVMQ*XVAL;
    PHWMVMQ = IHWMVMQ*XVAL;

    PLWHVMQ = ILWHVMQ*XVAL;
    PMWHVMQ = IMWHVMQ*XVAL;
    PHWHVMQ = IHWHVMQ*XVAL;

    PLWLVHQ = ILWLVHQ*XVAL;
    PMWLVHQ = IMWLVHQ*XVAL;
    PHWLVHQ = IHWLVHQ*XVAL;

    PLWMVHQ = ILWMVHQ*XVAL;
    PMWMVHQ = IMWMVHQ*XVAL;
    PHWMVHQ = IHWMVHQ*XVAL;

    PLWHVHQ = ILWHVHQ*XVAL;
    PMWHVHQ = IMWHVHQ*XVAL;
    PHWHVHQ = IHWHVHQ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 44: 1 or more mediators, in parallel if multiple (example uses 1), 3 moderators, 1 moderating both the IV- Mediator path and the direct IV-DV path, 2 moderating the Mediator-DV path with all 2-way and 3-way interactions, 1 of which also moderates the direct IV-DV path

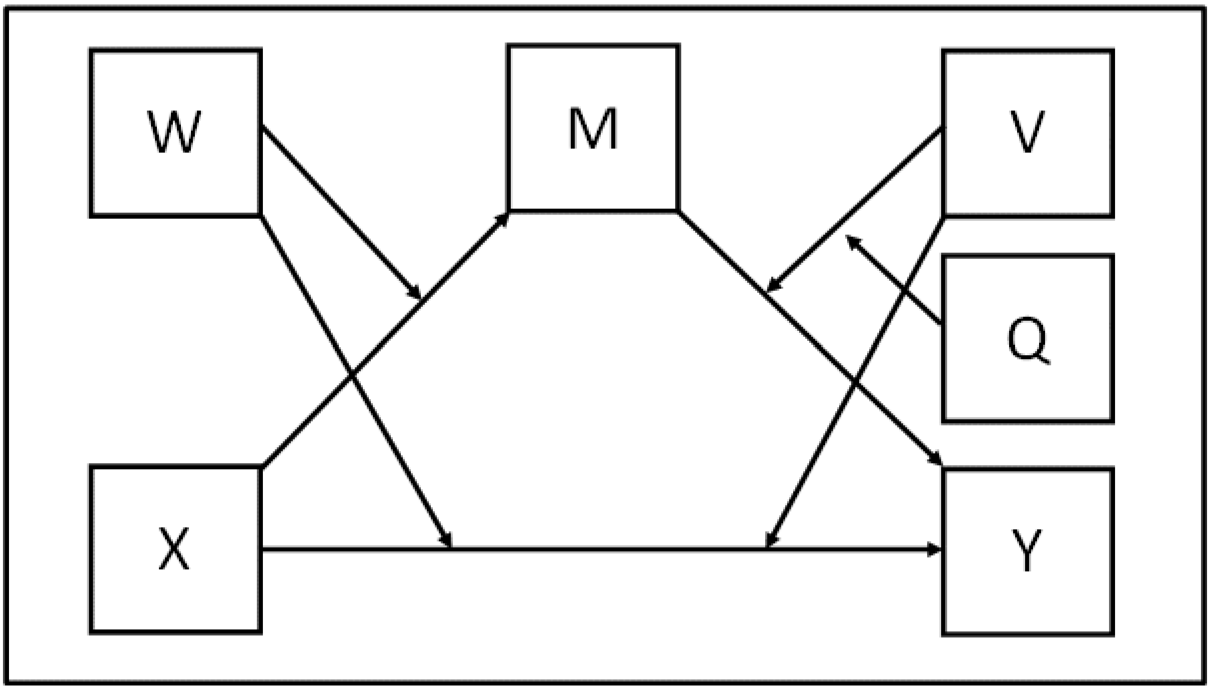
Example Variables: 1 predictor X, 1 mediator M, 3 moderators W, V, Q, 1 outcome Y

Preliminary notes:

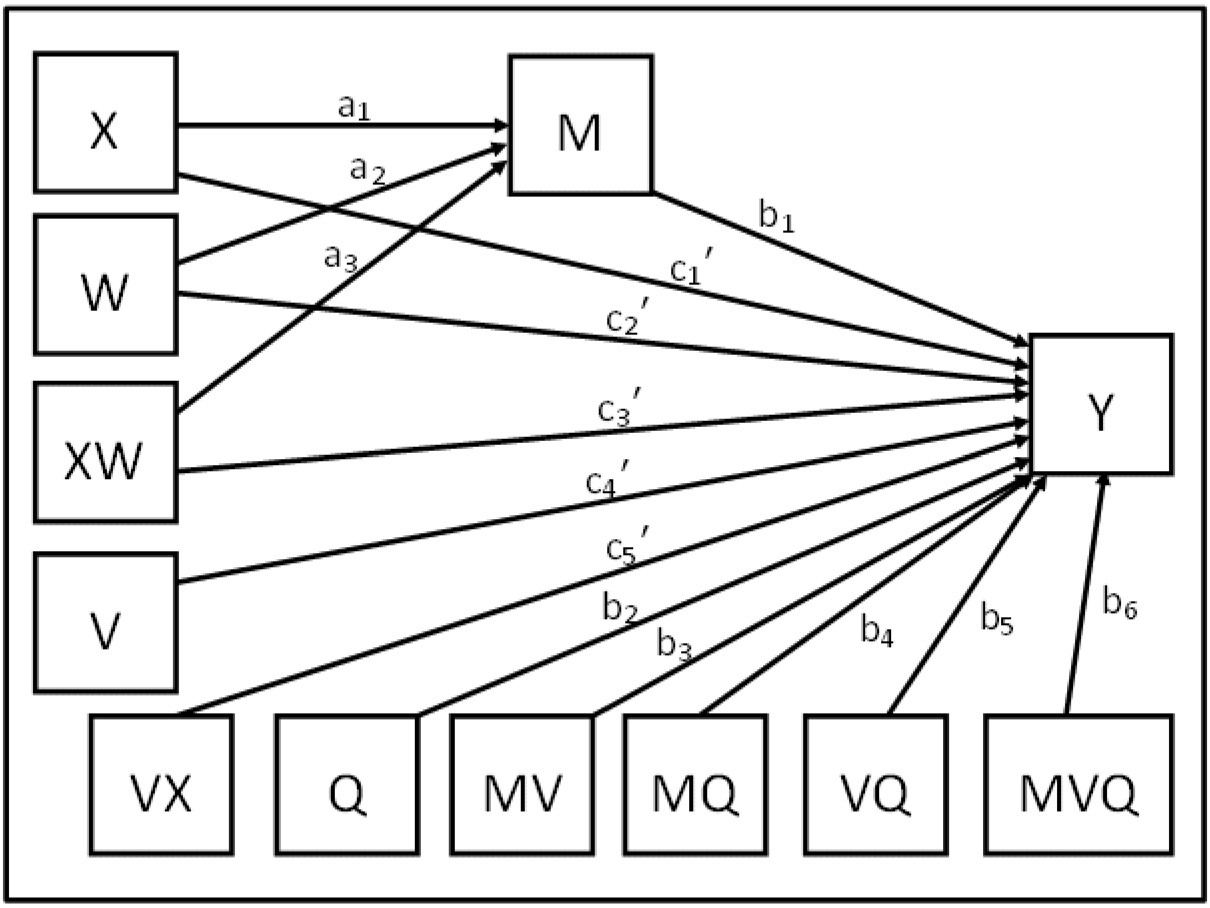
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2Q + b3MV + b4MQ + b5VQ + b6MVQ + c1'X + c2'W + c3'XW + c4'V + c5'XV
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2Q + b3MV + b4MQ + b5VQ + b6MVQ + c1'X + c2'W + c3'XW + c4'V + c5'XV
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2Q + b3(a0 + a1X + a2W + a3XW)V + b4(a0 + a1X + a2W + a3XW)Q + b5VQ + b6(a0 + a1X + a2W + a3XW)VQ + c1'X + c2'W + c3'XW + c4'V + c5'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2Q + a0b3V + a1b3XV + a2b3WV + a3b3XWV + a0b4Q + a1b4XQ + a2b4WQ + a3b4XWQ + b5VQ + a0b6VQ + a1b6XVQ + a2b6WVQ + a3b6XWVQ + c1'X + c2'W + c3'XW + c4'V + c5'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2Q + a0b3V + a2b3WV + a0b4Q + a2b4WQ + b5VQ + a0b6VQ + a2b6WVQ + c2'W + c4'V) + (a1b1 + a3b1W + a1b3V + a3b3WV + a1b4Q + a3b4WQ + a1b6VQ + a3b6WVQ + c1' + c3'W + c5'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V, Q:

a1b1 + a3b1W + a1b3V + a3b3WV + a1b4Q + a3b4WQ + a1b6VQ + a3b6WVQ = (a1 + a3W)(b1 + b3V + b4Q + b6VQ)

One direct effect of X on Y, conditional on W, V:

c1' + c3'W + c5'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V, Q
! Outcome variable - Y

USEVARIABLES = X M W V Q Y XW XV VQ MV MQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XV = X*V;
   VQ = V*Q;
   MVQ = M*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;
```

```
! In model statement name each path and intercept using
parentheses

MODEL:
    [Y] (b0);
   Y ON M (b1);
   Y ON Q (b2);
   Y ON MV (b3);
   Y ON MQ (b4);
   Y ON VQ (b5);
   Y ON MVQ (b6);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);
   Y ON V (cdash4);
   Y ON XV (cdash5);

    [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 3 moderators, 3 values for each, gives 27 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HWMVLQ = high value of W, medium value of V and low value of
Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V LOW_Q MED_Q
HIGH_Q
    ILWLVLQ IMWLVLQ IHWLVLQ ILWMVLQ IMWMVLQ IHWMVLQ
    ILWHVLQ IMWHVLQ IHWHVLQ
    ILWLVMQ IMWLVMQ IHWLVMQ ILWMVMQ IMWMVMQ IHWMVMQ
    ILWHVMQ IMWHVMQ IHWHVMQ
    ILWLVHQ IMWLVHQ IHWLVHQ ILWMVHQ IMWMVHQ IHWMVHQ
    ILWHVHQ IMWHVHQ IHWHVHQ
    DLOW_LOV DMEW_LOV DHIW_LOV DLOW_MEV DMEW_MEV DHIW_MEV
    DLOW_HIV DMEW_HIV DHIW_HIV
    TLWLVLQ TMWLVLQ THWLVLQ TLWMVLQ TMWMVLQ THWMVLQ
    TLWHVLQ TMWHVLQ THWHVLQ
    TLWLVMQ TMWLVMQ THWLVMQ TLWMVMQ TMWMVMQ THWMVMQ
    TLWHVMQ TMWHVMQ THWHVMQ
```

```
    TLWLVHQ TMWLVHQ THWLVHQ TLWMVHQ TMWMVHQ THWMVHQ
    TLWHVHQ TMWHVHQ THWHVHQ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILWLVLQ = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_V +
a3*b3*LOW_W*LOW_V +
     a1*b4*LOW_Q + a3*b4*LOW_W*LOW_Q + a1*b6*LOW_V*LOW_Q +
     a3*b6*LOW_W*LOW_V*LOW_Q;
    IMWLVLQ = a1*b1 + a3*b1*MED_W + a1*b3*LOW_V +
a3*b3*MED_W*LOW_V +
     a1*b4*LOW_Q + a3*b4*MED_W*LOW_Q + a1*b6*LOW_V*LOW_Q +
     a3*b6*MED_W*LOW_V*LOW_Q;
    IHWLVLQ = a1*b1 + a3*b1*HIGH_W + a1*b3*LOW_V +
a3*b3*HIGH_W*LOW_V +
     a1*b4*LOW_Q + a3*b4*HIGH_W*LOW_Q + a1*b6*LOW_V*LOW_Q +
     a3*b6*HIGH_W*LOW_V*LOW_Q;

    ILWMVLQ = a1*b1 + a3*b1*LOW_W + a1*b3*MED_V +
a3*b3*LOW_W*MED_V +
     a1*b4*LOW_Q + a3*b4*LOW_W*LOW_Q + a1*b6*MED_V*LOW_Q +
     a3*b6*LOW_W*MED_V*LOW_Q;
    IMWMVLQ = a1*b1 + a3*b1*MED_W + a1*b3*MED_V +
a3*b3*MED_W*MED_V +
     a1*b4*LOW_Q + a3*b4*MED_W*LOW_Q + a1*b6*MED_V*LOW_Q +
     a3*b6*MED_W*MED_V*LOW_Q;
    IHWMVLQ = a1*b1 + a3*b1*HIGH_W + a1*b3*MED_V +
a3*b3*HIGH_W*MED_V +
```

```
     a1*b4*LOW_Q + a3*b4*HIGH_W*LOW_Q + a1*b6*MED_V*LOW_Q +
     a3*b6*HIGH_W*MED_V*LOW_Q;

    ILWHVLQ = a1*b1 + a3*b1*LOW_W + a1*b3*HIGH_V +
a3*b3*LOW_W*HIGH_V +
     a1*b4*LOW_Q + a3*b4*LOW_W*LOW_Q + a1*b6*HIGH_V*LOW_Q +
     a3*b6*LOW_W*HIGH_V*LOW_Q;
    IMWHVLQ = a1*b1 + a3*b1*MED_W + a1*b3*HIGH_V +
a3*b3*MED_W*HIGH_V +
     a1*b4*LOW_Q + a3*b4*MED_W*LOW_Q + a1*b6*HIGH_V*LOW_Q +
     a3*b6*MED_W*HIGH_V*LOW_Q;
    IHWHVLQ = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_V +
a3*b3*HIGH_W*HIGH_V +
     a1*b4*LOW_Q + a3*b4*HIGH_W*LOW_Q + a1*b6*HIGH_V*LOW_Q +
     a3*b6*HIGH_W*HIGH_V*LOW_Q;

    ILWLVMQ = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_V +
a3*b3*LOW_W*LOW_V +
     a1*b4*MED_Q + a3*b4*LOW_W*MED_Q + a1*b6*LOW_V*MED_Q +
     a3*b6*LOW_W*LOW_V*MED_Q;
    IMWLVMQ = a1*b1 + a3*b1*MED_W + a1*b3*LOW_V +
a3*b3*MED_W*LOW_V +
     a1*b4*MED_Q + a3*b4*MED_W*MED_Q + a1*b6*LOW_V*MED_Q +
     a3*b6*MED_W*LOW_V*MED_Q;
    IHWLVMQ = a1*b1 + a3*b1*HIGH_W + a1*b3*LOW_V +
a3*b3*HIGH_W*LOW_V +
     a1*b4*MED_Q + a3*b4*HIGH_W*MED_Q + a1*b6*LOW_V*MED_Q +
     a3*b6*HIGH_W*LOW_V*MED_Q;

    ILWMVMQ = a1*b1 + a3*b1*LOW_W + a1*b3*MED_V +
a3*b3*LOW_W*MED_V +
     a1*b4*MED_Q + a3*b4*LOW_W*MED_Q + a1*b6*MED_V*MED_Q +
     a3*b6*LOW_W*MED_V*MED_Q;
    IMWMVMQ = a1*b1 + a3*b1*MED_W + a1*b3*MED_V +
a3*b3*MED_W*MED_V +
     a1*b4*MED_Q + a3*b4*MED_W*MED_Q + a1*b6*MED_V*MED_Q +
     a3*b6*MED_W*MED_V*MED_Q;
    IHWMVMQ = a1*b1 + a3*b1*HIGH_W + a1*b3*MED_V +
a3*b3*HIGH_W*MED_V +
     a1*b4*MED_Q + a3*b4*HIGH_W*MED_Q + a1*b6*MED_V*MED_Q +
     a3*b6*HIGH_W*MED_V*MED_Q;

    ILWHVMQ = a1*b1 + a3*b1*LOW_W + a1*b3*HIGH_V +
a3*b3*LOW_W*HIGH_V +
     a1*b4*MED_Q + a3*b4*LOW_W*MED_Q + a1*b6*HIGH_V*MED_Q +
     a3*b6*LOW_W*HIGH_V*MED_Q;
    IMWHVMQ = a1*b1 + a3*b1*MED_W + a1*b3*HIGH_V +
a3*b3*MED_W*HIGH_V +
     a1*b4*MED_Q + a3*b4*MED_W*MED_Q + a1*b6*HIGH_V*MED_Q +
```

```
     a3*b6*MED_W*HIGH_V*MED_Q;
    IHWHVMQ = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_V +
a3*b3*HIGH_W*HIGH_V +
     a1*b4*MED_Q + a3*b4*HIGH_W*MED_Q + a1*b6*HIGH_V*MED_Q +
     a3*b6*HIGH_W*HIGH_V*MED_Q;

    ILWLVHQ = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_V +
a3*b3*LOW_W*LOW_V +
     a1*b4*HIGH_Q + a3*b4*LOW_W*HIGH_Q + a1*b6*LOW_V*HIGH_Q +
     a3*b6*LOW_W*LOW_V*HIGH_Q;
    IMWLVHQ = a1*b1 + a3*b1*MED_W + a1*b3*LOW_V +
a3*b3*MED_W*LOW_V +
     a1*b4*HIGH_Q + a3*b4*MED_W*HIGH_Q + a1*b6*LOW_V*HIGH_Q +
     a3*b6*MED_W*LOW_V*HIGH_Q;
    IHWLVHQ = a1*b1 + a3*b1*HIGH_W + a1*b3*LOW_V +
a3*b3*HIGH_W*LOW_V +
     a1*b4*HIGH_Q + a3*b4*HIGH_W*HIGH_Q + a1*b6*LOW_V*HIGH_Q
+
     a3*b6*HIGH_W*LOW_V*HIGH_Q;

    ILWMVHQ = a1*b1 + a3*b1*LOW_W + a1*b3*MED_V +
a3*b3*LOW_W*MED_V +
     a1*b4*HIGH_Q + a3*b4*LOW_W*HIGH_Q + a1*b6*MED_V*HIGH_Q +
     a3*b6*LOW_W*MED_V*HIGH_Q;
    IMWMVHQ = a1*b1 + a3*b1*MED_W + a1*b3*MED_V +
a3*b3*MED_W*MED_V +
     a1*b4*HIGH_Q + a3*b4*MED_W*HIGH_Q + a1*b6*MED_V*HIGH_Q +
     a3*b6*MED_W*MED_V*HIGH_Q;
    IHWMVHQ = a1*b1 + a3*b1*HIGH_W + a1*b3*MED_V +
a3*b3*HIGH_W*MED_V +
     a1*b4*HIGH_Q + a3*b4*HIGH_W*HIGH_Q + a1*b6*MED_V*HIGH_Q
+
     a3*b6*HIGH_W*MED_V*HIGH_Q;

    ILWHVHQ = a1*b1 + a3*b1*LOW_W + a1*b3*HIGH_V +
a3*b3*LOW_W*HIGH_V +
     a1*b4*HIGH_Q + a3*b4*LOW_W*HIGH_Q + a1*b6*HIGH_V*HIGH_Q
+
     a3*b6*LOW_W*HIGH_V*HIGH_Q;
    IMWHVHQ = a1*b1 + a3*b1*MED_W + a1*b3*HIGH_V +
a3*b3*MED_W*HIGH_V +
     a1*b4*HIGH_Q + a3*b4*MED_W*HIGH_Q + a1*b6*HIGH_V*HIGH_Q
+
     a3*b6*MED_W*HIGH_V*HIGH_Q;
    IHWHVHQ = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_V +
a3*b3*HIGH_W*HIGH_V +
     a1*b4*HIGH_Q + a3*b4*HIGH_W*HIGH_Q + a1*b6*HIGH_V*HIGH_Q
+
     a3*b6*HIGH_W*HIGH_V*HIGH_Q;
```

```
! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOV = cdash1 + cdash3*LOW_W + cdash5*LOW_V;
    DMEW_LOV = cdash1 + cdash3*MED_W + cdash5*LOW_V;
    DHIW_LOV = cdash1 + cdash3*HIGH_W + cdash5*LOW_V;

    DLOW_MEV = cdash1 + cdash3*LOW_W + cdash5*MED_V;
    DMEW_MEV = cdash1 + cdash3*MED_W + cdash5*MED_V;
    DHIW_MEV = cdash1 + cdash3*HIGH_W + cdash5*MED_V;

    DLOW_HIV = cdash1 + cdash3*LOW_W + cdash5*HIGH_V;
    DMEW_HIV = cdash1 + cdash3*MED_W + cdash5*HIGH_V;
    DHIW_HIV = cdash1 + cdash3*HIGH_W + cdash5*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLWLVLQ = ILWLVLQ + DLOW_LOV;
    TMWLVLQ = IMWLVLQ + DMEW_LOV;
    THWLVLQ = IHWLVLQ + DHIW_LOV;

    TLWMVLQ = ILWMVLQ + DLOW_MEV;
    TMWMVLQ = IMWMVLQ + DMEW_MEV;
    THWMVLQ = IHWMVLQ + DHIW_MEV;

    TLWHVLQ = ILWHVLQ + DLOW_HIV;
    TMWHVLQ = IMWHVLQ + DMEW_HIV;
    THWHVLQ = IHWHVLQ + DHIW_HIV;

    TLWLVMQ = ILWLVMQ + DLOW_LOV;
    TMWLVMQ = IMWLVMQ + DMEW_LOV;
    THWLVMQ = IHWLVMQ + DHIW_LOV;

    TLWMVMQ = ILWMVMQ + DLOW_MEV;
    TMWMVMQ = IMWMVMQ + DMEW_MEV;
    THWMVMQ = IHWMVMQ + DHIW_MEV;

    TLWHVMQ = ILWHVMQ + DLOW_HIV;
    TMWHVMQ = IMWHVMQ + DMEW_HIV;
    THWHVMQ = IHWHVMQ + DHIW_HIV;

    TLWLVHQ = ILWLVHQ + DLOW_LOV;
    TMWLVHQ = IMWLVHQ + DMEW_LOV;
    THWLVHQ = IHWLVHQ + DHIW_LOV;

    TLWMVHQ = ILWMVHQ + DLOW_MEV;
    TMWMVHQ = IMWMVHQ + DMEW_MEV;
    THWMVHQ = IHWMVHQ + DHIW_MEV;

    TLWHVHQ = ILWHVHQ + DLOW_HIV;
    TMWHVHQ = IMWHVHQ + DMEW_HIV;
    THWHVHQ = IHWHVHQ + DHIW_HIV;
```

```
! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLWLVLQ PMWLVLQ PHWLVLQ PLWMVLQ PMWMVLQ PHWMVLQ
    PLWHVLQ PMWHVLQ PHWHVLQ
    PLWLVMQ PMWLVMQ PHWLVMQ PLWMVMQ PMWMVMQ PHWMVMQ
    PLWHVMQ PMWHVMQ PHWHVMQ
    PLWLVHQ PMWLVHQ PHWLVHQ PLWMVHQ PMWMVHQ PHWMVHQ
    PLWHVHQ PMWHVHQ PHWHVHQ);

    LOOP(XVAL,1,5,0.1);

    PLWLVLQ = ILWLVLQ*XVAL;
    PMWLVLQ = IMWLVLQ*XVAL;
    PHWLVLQ = IHWLVLQ*XVAL;

    PLWMVLQ = ILWMVLQ*XVAL;
    PMWMVLQ = IMWMVLQ*XVAL;
    PHWMVLQ = IHWMVLQ*XVAL;

    PLWHVLQ = ILWHVLQ*XVAL;
    PMWHVLQ = IMWHVLQ*XVAL;
    PHWHVLQ = IHWHVLQ*XVAL;

    PLWLVMQ = ILWLVMQ*XVAL;
    PMWLVMQ = IMWLVMQ*XVAL;
    PHWLVMQ = IHWLVMQ*XVAL;

    PLWMVMQ = ILWMVMQ*XVAL;
    PMWMVMQ = IMWMVMQ*XVAL;
    PHWMVMQ = IHWMVMQ*XVAL;

    PLWHVMQ = ILWHVMQ*XVAL;
    PMWHVMQ = IMWHVMQ*XVAL;
    PHWHVMQ = IHWHVMQ*XVAL;

    PLWLVHQ = ILWLVHQ*XVAL;
    PMWLVHQ = IMWLVHQ*XVAL;
    PHWLVHQ = IHWLVHQ*XVAL;

    PLWMVHQ = ILWMVHQ*XVAL;
    PMWMVHQ = IMWMVHQ*XVAL;
    PHWMVHQ = IHWMVHQ*XVAL;

    PLWHVHQ = ILWHVHQ*XVAL;
    PMWHVHQ = IMWHVHQ*XVAL;
    PHWHVHQ = IHWHVHQ*XVAL;
```

```
PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 45: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path, with the other 2 moderating the Mediator-DV path

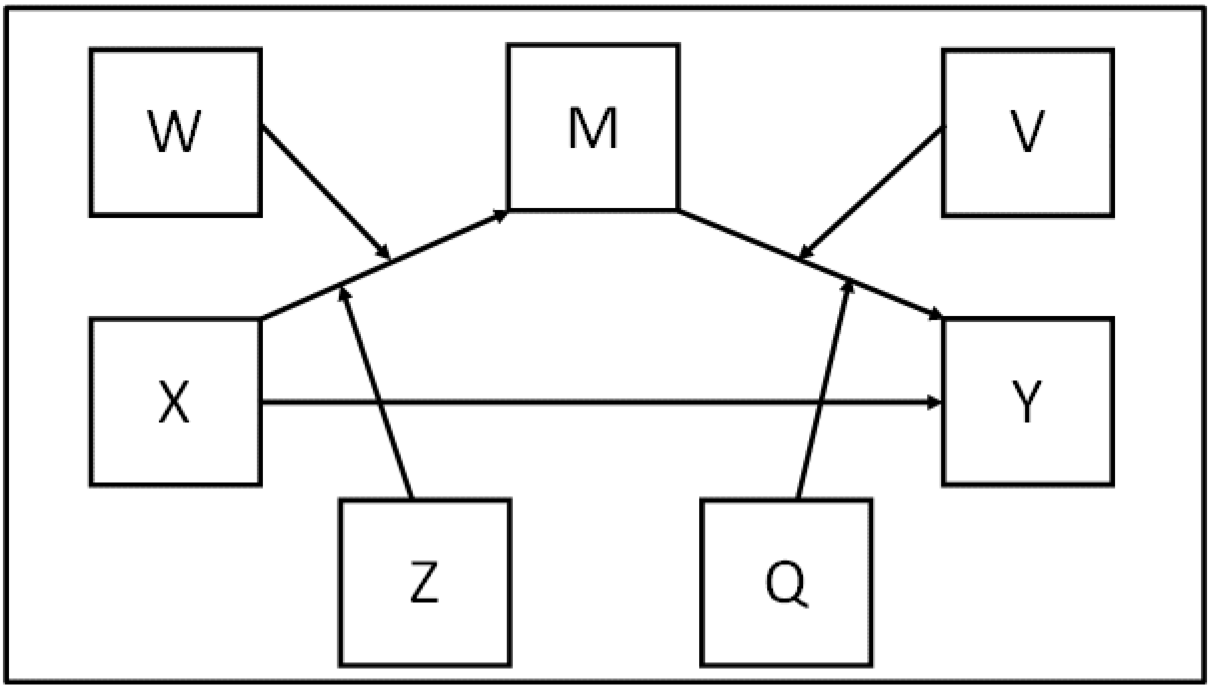
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

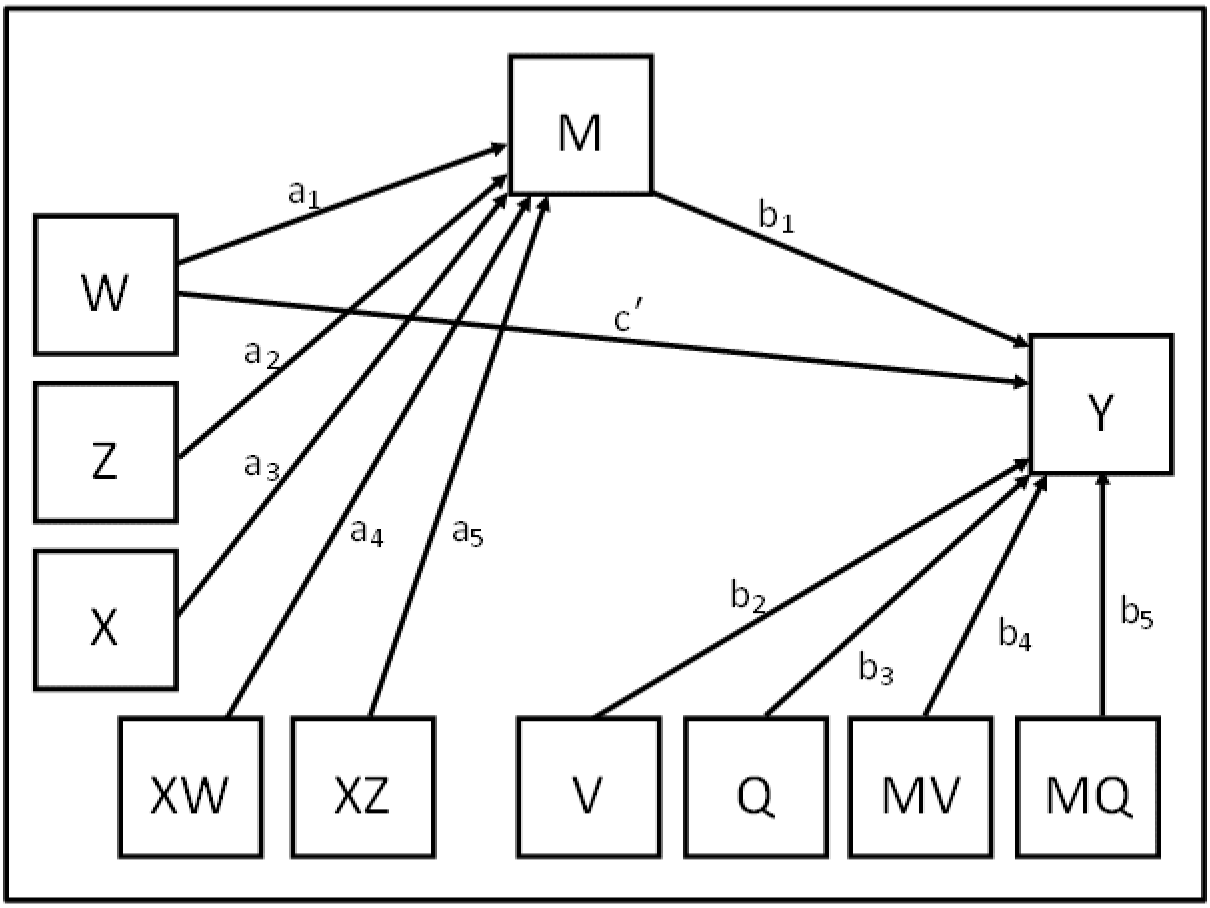
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

$Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c'X$
$M = a0 + a1W + a2Z + a3X + a4XW + a5XZ$

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

$Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c'X$
$M = a0 + a1W + a2Z + a3X + a4XW + a5XZ$

Hence... substituting in equation for M

$Y = b0 + b1(a0 + a1W + a2Z + a3X + a4XW + a5XZ) + b2V + b3Q + b4(a0 + a1W + a2Z + a3X + a4XW + a5XZ)V + b5(a0 + a1W + a2Z + a3X + a4XW + a5XZ)Q + c'X$

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1W + a2b1Z + a3b1X + a4b1XW + a5b1XZ + b2V + b3Q + a0b4V + a1b4WV + a2b4ZV + a3b4XV + a4b4XWV + a5b4XZV + a0b5Q + a1b5WQ + a2b5ZQ + a3b5XQ + a4b5XWQ + a5b5XZQ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a1b1W + a2b1Z + b2V + b3Q + a0b4V + a1b4WV + a2b4ZV + a0b5Q + a1b5WQ + a2b5ZQ) + (a3b1 + a4b1W + a5b1Z + a3b4V + a4b4WV + a5b4ZV + a3b5Q + a4b5WQ + a5b5ZQ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a3b1 + a4b1W + a5b1Z + a3b4V + a4b4WV + a5b4ZV + a3b5Q + a4b5WQ + a5b5ZQ = (a3 + a4W + a5Z)(b1 + b4V + b5Q)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ MV MQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
```

```
    Y ON M (b1);
    Y ON V (b2);
    Y ON Q (b3);
    Y ON MV (b4);
    Y ON MQ (b5);

    Y ON X (cdash);

    [M] (a0);
    M ON W (a1);
    M ON Z (a2);
    M ON X (a3);
    M ON XW (a4);
    M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);
```

```
    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a3*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a3*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a3*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
    IMLLL = a3*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a3*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a3*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
    IHLLL = a3*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a3*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a3*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;

    ILMLL = a3*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a3*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a3*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
    IMMLL = a3*b1 + a4*b1*MED_W + a5*b1*MED_Z + a3*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a3*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
    IHMLL = a3*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a3*b4*LOW_V
+
```

```
 a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a3*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q;

ILHLL = a3*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a3*b4*LOW_V
+
 a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a3*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IMHLL = a3*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a3*b4*LOW_V
+
 a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a3*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IHHLL = a3*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a3*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a3*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;

ILLML = a3*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a3*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a3*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
IMLML = a3*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a3*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a3*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
IHLML = a3*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a3*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a3*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;

ILMML = a3*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a3*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a3*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
IMMML = a3*b1 + a4*b1*MED_W + a5*b1*MED_Z + a3*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a3*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
IHMML = a3*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a3*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a3*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q;

ILHML = a3*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a3*b4*MED_V
+
 a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a3*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IMHML = a3*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a3*b4*MED_V
+
 a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a3*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IHHML = a3*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a3*b4*MED_V
+
```

```
 a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a3*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;

ILLHL = a3*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a3*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a3*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
IMLHL = a3*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a3*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a3*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
IHLHL = a3*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a3*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a3*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;

ILMHL = a3*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a3*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a3*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
IMMHL = a3*b1 + a4*b1*MED_W + a5*b1*MED_Z + a3*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a3*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
IHMHL = a3*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a3*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a3*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q;

ILHHL = a3*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a3*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a3*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IMHHL = a3*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a3*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a3*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IHHHL = a3*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a3*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a3*b5*LOW_Q
+
 a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;

ILLLM = a3*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a3*b4*LOW_V +
 a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a3*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q;
IMLLM = a3*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a3*b4*LOW_V +
 a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a3*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q;
```

```
    IHLLM = a3*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a3*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a3*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q;
    ILMLM = a3*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a3*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a3*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q;
    IMMLM = a3*b1 + a4*b1*MED_W + a5*b1*MED_Z + a3*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a3*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q;
    IHMLM = a3*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a3*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a3*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q;
    ILHLM = a3*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a3*b4*LOW_V
+
     a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a3*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
    IMHLM = a3*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a3*b4*LOW_V
+
     a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a3*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
    IHHLM = a3*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a3*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a3*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
    ILLMM = a3*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a3*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a3*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q;
    IMLMM = a3*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a3*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a3*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q;
    IHLMM = a3*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a3*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a3*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q;
    ILMMM = a3*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a3*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a3*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q;
    IMMMM = a3*b1 + a4*b1*MED_W + a5*b1*MED_Z + a3*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a3*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q;
    IHMMM = a3*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a3*b4*MED_V
+
```

```
 a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a3*b5*MED_Q +
 a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q;

ILHMM = a3*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a3*b4*MED_V
+
 a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a3*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
IMHMM = a3*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a3*b4*MED_V
+
 a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a3*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
IHHMM = a3*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a3*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a3*b5*MED_Q +
 a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q;

ILLHM = a3*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a3*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a3*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q;
IMLHM = a3*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a3*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a3*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q;
IHLHM = a3*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a3*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a3*b5*MED_Q +
 a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q;

ILMHM = a3*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a3*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a3*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q;
IMMHM = a3*b1 + a4*b1*MED_W + a5*b1*MED_Z + a3*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a3*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q;
IHMHM = a3*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a3*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a3*b5*MED_Q +
 a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q;

ILHHM = a3*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a3*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a3*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
IMHHM = a3*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a3*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a3*b5*MED_Q +
```

```
 a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
IHHHM = a3*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a3*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a3*b5*MED_Q
+
 a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q;

ILLLH = a3*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a3*b4*LOW_V +
 a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a3*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IMLLH = a3*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a3*b4*LOW_V +
 a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a3*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IHLLH = a3*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a3*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a3*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;

ILMLH = a3*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a3*b4*LOW_V +
 a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a3*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IMMLH = a3*b1 + a4*b1*MED_W + a5*b1*MED_Z + a3*b4*LOW_V +
 a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a3*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IHMLH = a3*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a3*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a3*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;

ILHLH = a3*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a3*b4*LOW_V
+
 a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a3*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
IMHLH = a3*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a3*b4*LOW_V
+
 a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a3*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
IHHLH = a3*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a3*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a3*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;

ILLMH = a3*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a3*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a3*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IMLMH = a3*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a3*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a3*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IHLMH = a3*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a3*b4*MED_V
```

```
+
 a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a3*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;

ILMMH = a3*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a3*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a3*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IMMMH = a3*b1 + a4*b1*MED_W + a5*b1*MED_Z + a3*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a3*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IHMMH = a3*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a3*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a3*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;

ILHMH = a3*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a3*b4*MED_V
+
 a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a3*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
IMHMH = a3*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a3*b4*MED_V
+
 a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a3*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
IHHMH = a3*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a3*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a3*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;

ILLHH = a3*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a3*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a3*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IMLHH = a3*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a3*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a3*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IHLHH = a3*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a3*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a3*b5*HIGH_Q
+
 a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;

ILMHH = a3*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a3*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a3*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IMMHH = a3*b1 + a4*b1*MED_W + a5*b1*MED_Z + a3*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a3*b5*HIGH_Q +
```

```
     a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
    IHMHH = a3*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a3*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a3*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;

    ILHHH = a3*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a3*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a3*b5*HIGH_Q
+
     a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
    IMHHH = a3*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a3*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a3*b5*HIGH_Q
+
     a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
    IHHHH = a3*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a3*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a3*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + cdash;
    TMLLL = IMLLL + cdash;
    THLLL = IHLLL + cdash;

    TLMLL = ILMLL + cdash;
    TMMLL = IMMLL + cdash;
    THMLL = IHMLL + cdash;

    TLHLL = ILHLL + cdash;
    TMHLL = IMHLL + cdash;
    THHLL = IHHLL + cdash;

    TLLML = ILLML + cdash;
    TMLML = IMLML + cdash;
    THLML = IHLML + cdash;

    TLMML = ILMML + cdash;
    TMMML = IMMML + cdash;
    THMML = IHMML + cdash;

    TLHML = ILHML + cdash;
    TMHML = IMHML + cdash;
    THHML = IHHML + cdash;
```

```
TLLHL = ILLHL + cdash;
TMLHL = IMLHL + cdash;
THLHL = IHLHL + cdash;

TLMHL = ILMHL + cdash;
TMMHL = IMMHL + cdash;
THMHL = IHMHL + cdash;

TLHHL = ILHHL + cdash;
TMHHL = IMHHL + cdash;
THHHL = IHHHL + cdash;

TLLLM = ILLLM + cdash;
TMLLM = IMLLM + cdash;
THLLM = IHLLM + cdash;

TLMLM = ILMLM + cdash;
TMMLM = IMMLM + cdash;
THMLM = IHMLM + cdash;

TLHLM = ILHLM + cdash;
TMHLM = IMHLM + cdash;
THHLM = IHHLM + cdash;

TLLMM = ILLMM + cdash;
TMLMM = IMLMM + cdash;
THLMM = IHLMM + cdash;

TLMMM = ILMMM + cdash;
TMMMM = IMMMM + cdash;
THMMM = IHMMM + cdash;

TLHMM = ILHMM + cdash;
TMHMM = IMHMM + cdash;
THHMM = IHHMM + cdash;

TLLHM = ILLHM + cdash;
TMLHM = IMLHM + cdash;
THLHM = IHLHM + cdash;

TLMHM = ILMHM + cdash;
TMMHM = IMMHM + cdash;
THMHM = IHMHM + cdash;

TLHHM = ILHHM + cdash;
TMHHM = IMHHM + cdash;
THHHM = IHHHM + cdash;

TLLLH = ILLLH + cdash;
TMLLH = IMLLH + cdash;
THLLH = IHLLH + cdash;
```

```
    TLMLH = ILMLH + cdash;
    TMMLH = IMMLH + cdash;
    THMLH = IHMLH + cdash;

    TLHLH = ILHLH + cdash;
    TMHLH = IMHLH + cdash;
    THHLH = IHHLH + cdash;

    TLLMH = ILLMH + cdash;
    TMLMH = IMLMH + cdash;
    THLMH = IHLMH + cdash;

    TLMMH = ILMMH + cdash;
    TMMMH = IMMMH + cdash;
    THMMH = IHMMH + cdash;

    TLHMH = ILHMH + cdash;
    TMHMH = IMHMH + cdash;
    THHMH = IHHMH + cdash;

    TLLHH = ILLHH + cdash;
    TMLHH = IMLHH + cdash;
    THLHH = IHLHH + cdash;

    TLMHH = ILMHH + cdash;
    TMMHH = IMMHH + cdash;
    THMHH = IHMHH + cdash;

    TLHHH = ILHHH + cdash;
    TMHHH = IMHHH + cdash;
    THHHH = IHHHH + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);
```

```
PLLLL = ILLLL*XVAL;
PMLLL = IMLLL*XVAL;
PHLLL = IHLLL*XVAL;

PLMLL = ILMLL*XVAL;
PMMLL = IMMLL*XVAL;
PHMLL = IHMLL*XVAL;

PLHLL = ILHLL*XVAL;
PMHLL = IMHLL*XVAL;
PHHLL = IHHLL*XVAL;

PLLML = ILLML*XVAL;
PMLML = IMLML*XVAL;
PHLML = IHLML*XVAL;

PLMML = ILMML*XVAL;
PMMML = IMMML*XVAL;
PHMML = IHMML*XVAL;

PLHML = ILHML*XVAL;
PMHML = IMHML*XVAL;
PHHML = IHHML*XVAL;

PLLHL = ILLHL*XVAL;
PMLHL = IMLHL*XVAL;
PHLHL = IHLHL*XVAL;

PLMHL = ILMHL*XVAL;
PMMHL = IMMHL*XVAL;
PHMHL = IHMHL*XVAL;

PLHHL = ILHHL*XVAL;
PMHHL = IMHHL*XVAL;
PHHHL = IHHHL*XVAL;

PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;
```

```
PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;

PLHLH = ILHLH*XVAL;
PMHLH = IMHLH*XVAL;
PHHLH = IHHLH*XVAL;

PLLMH = ILLMH*XVAL;
PMLMH = IMLMH*XVAL;
PHLMH = IHLMH*XVAL;

PLMMH = ILMMH*XVAL;
PMMMH = IMMMH*XVAL;
PHMMH = IHMMH*XVAL;

PLHMH = ILHMH*XVAL;
PMHMH = IMHMH*XVAL;
PHHMH = IHHMH*XVAL;

PLLHH = ILLHH*XVAL;
PMLHH = IMLHH*XVAL;
PHLHH = IHLHH*XVAL;

PLMHH = ILMHH*XVAL;
PMMHH = IMMHH*XVAL;
PHMHH = IHMHH*XVAL;
```

```
    PLHHH = ILHHH*XVAL;
    PMHHH = IMHHH*XVAL;
    PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 46: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path with all 2-way and 3-way interactions, with the other 2 moderating the Mediator-DV path

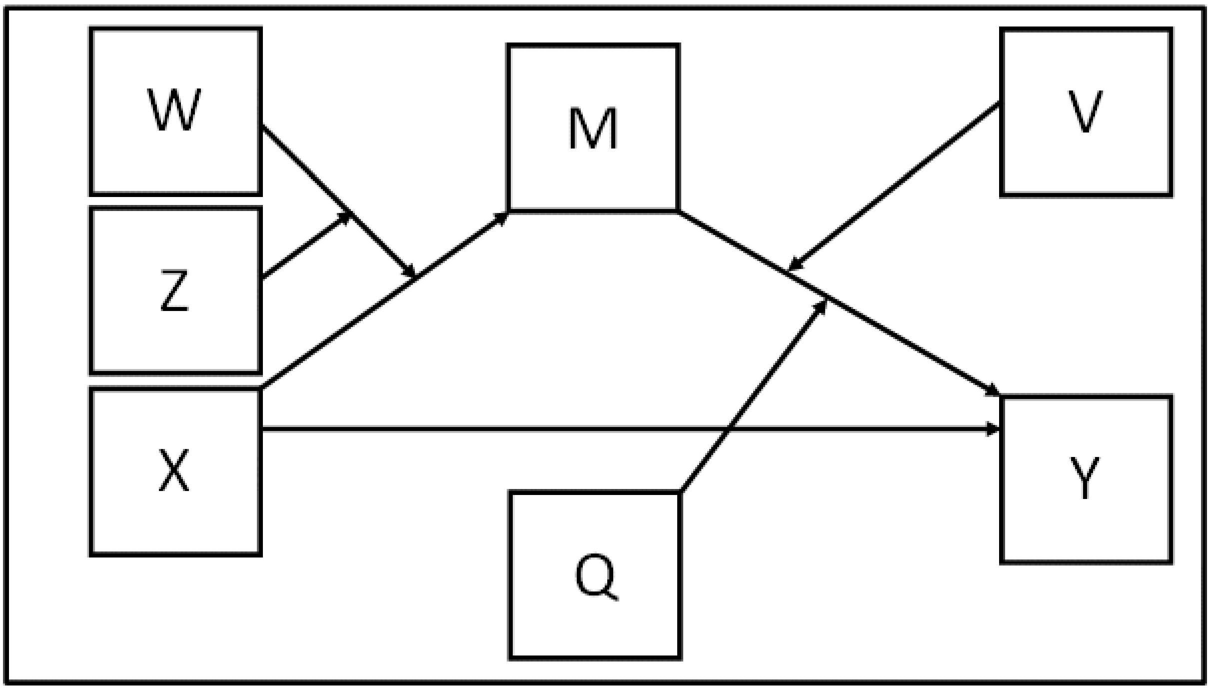
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

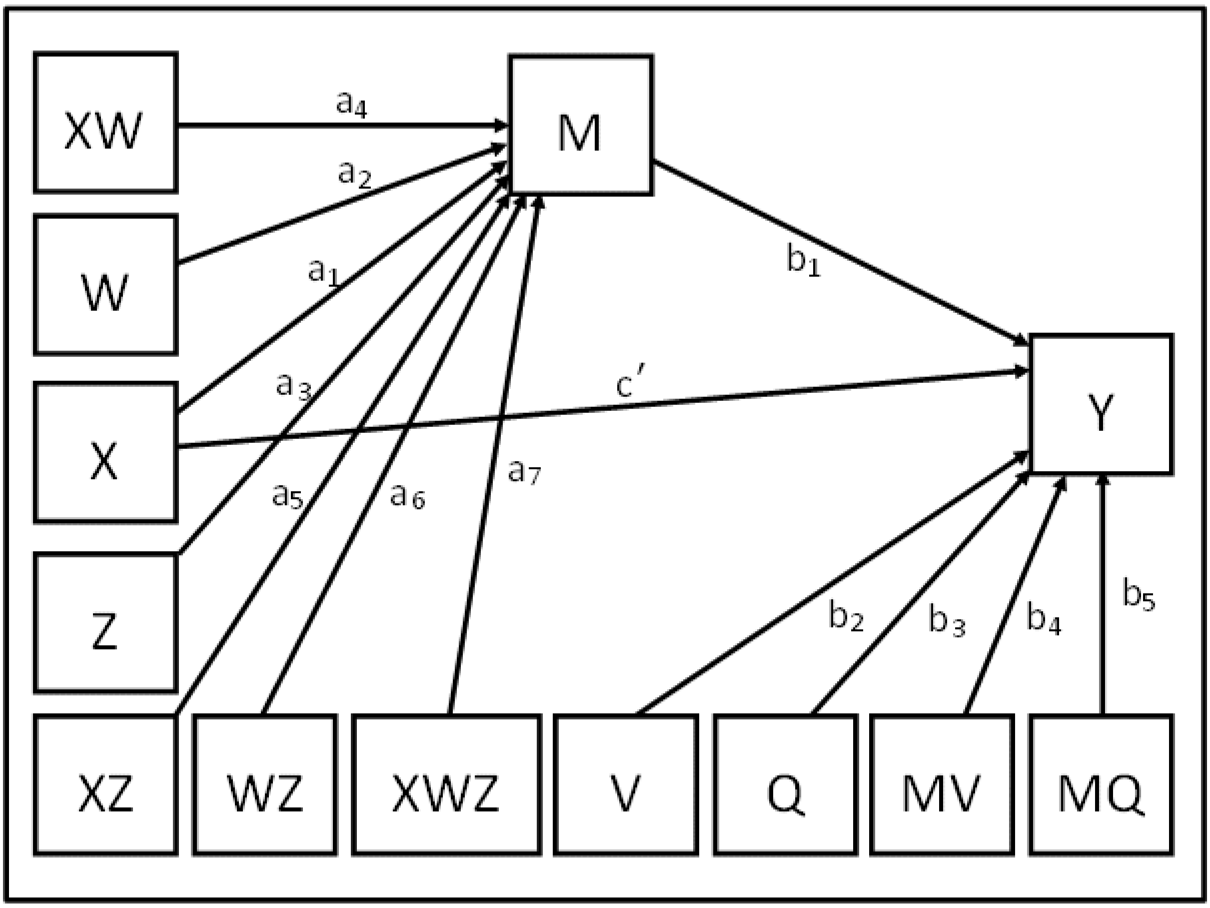
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2V + b3Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + b5(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)Q + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4ZV + a4b4XWV + a5b4XZV + a6b4WZV + a7b4XWZV + a0b5Q + a1b5XQ + a2b5WQ + a3b5ZQ + a4b5XWQ + a5b5XZQ + a6b5WZQ + a7b5XWZQ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + b2V + b3Q + a0b4V + a2b4WV + a3b4ZV + a6b4WZV + a0b5Q + a2b5WQ + a3b5ZQ + a6b5WZQ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4V + a4b4WV + a5b4ZV + a7b4WZV + a1b5Q + a4b5WQ + a5b5ZQ + a7b5WZQ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4V + a4b4WV + a5b4ZV + a7b4WZV + a1b5Q + a4b5WQ + a5b5ZQ + a7b5WZQ = (a1 + a4W + a5Z + a7WZ)(b1 + b4V + b5Q)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ WZ MV MQ XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;
```

```
! In model statement name each path and intercept using
parentheses

MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON V (b2);
    Y ON Q (b3);
    Y ON MV (b4);
    Y ON MQ (b5);

    Y ON X (cdash);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
    M ON WZ (a6);
    M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
```

```
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
```

```
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q;

    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q;

    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
```

```
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q;
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
```

```
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q;

    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
```

```
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q;
```

```
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q;

    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q;

    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q;
    IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q;

    ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
```

```
+
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q;
    IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q;
    IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q;
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
```

```
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q;
```

```
    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
```

```
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q;
    IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q;

    ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q;
    IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q;
    IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q;

    ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q;
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
```

```
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q;

    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
```

```
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + cdash;
    TMLLL = IMLLL + cdash;
    THLLL = IHLLL + cdash;

    TLMLL = ILMLL + cdash;
    TMMLL = IMMLL + cdash;
    THMLL = IHMLL + cdash;

    TLHLL = ILHLL + cdash;
    TMHLL = IMHLL + cdash;
    THHLL = IHHLL + cdash;

    TLLML = ILLML + cdash;
    TMLML = IMLML + cdash;
    THLML = IHLML + cdash;
```

```
TLMML = ILMML + cdash;
TMMML = IMMML + cdash;
THMML = IHMML + cdash;

TLHML = ILHML + cdash;
TMHML = IMHML + cdash;
THHML = IHHML + cdash;

TLLHL = ILLHL + cdash;
TMLHL = IMLHL + cdash;
THLHL = IHLHL + cdash;

TLMHL = ILMHL + cdash;
TMMHL = IMMHL + cdash;
THMHL = IHMHL + cdash;

TLHHL = ILHHL + cdash;
TMHHL = IMHHL + cdash;
THHHL = IHHHL + cdash;

TLLLM = ILLLM + cdash;
TMLLM = IMLLM + cdash;
THLLM = IHLLM + cdash;

TLMLM = ILMLM + cdash;
TMMLM = IMMLM + cdash;
THMLM = IHMLM + cdash;

TLHLM = ILHLM + cdash;
TMHLM = IMHLM + cdash;
THHLM = IHHLM + cdash;

TLLMM = ILLMM + cdash;
TMLMM = IMLMM + cdash;
THLMM = IHLMM + cdash;

TLMMM = ILMMM + cdash;
TMMMM = IMMMM + cdash;
THMMM = IHMMM + cdash;

TLHMM = ILHMM + cdash;
TMHMM = IMHMM + cdash;
THHMM = IHHMM + cdash;

TLLHM = ILLHM + cdash;
TMLHM = IMLHM + cdash;
THLHM = IHLHM + cdash;

TLMHM = ILMHM + cdash;
TMMHM = IMMHM + cdash;
THMHM = IHMHM + cdash;
```

```
    TLHHM = ILHHM + cdash;
    TMHHM = IMHHM + cdash;
    THHHM = IHHHM + cdash;

    TLLLH = ILLLH + cdash;
    TMLLH = IMLLH + cdash;
    THLLH = IHLLH + cdash;

    TLMLH = ILMLH + cdash;
    TMMLH = IMMLH + cdash;
    THMLH = IHMLH + cdash;

    TLHLH = ILHLH + cdash;
    TMHLH = IMHLH + cdash;
    THHLH = IHHLH + cdash;

    TLLMH = ILLMH + cdash;
    TMLMH = IMLMH + cdash;
    THLMH = IHLMH + cdash;

    TLMMH = ILMMH + cdash;
    TMMMH = IMMMH + cdash;
    THMMH = IHMMH + cdash;

    TLHMH = ILHMH + cdash;
    TMHMH = IMHMH + cdash;
    THHMH = IHHMH + cdash;

    TLLHH = ILLHH + cdash;
    TMLHH = IMLHH + cdash;
    THLHH = IHLHH + cdash;

    TLMHH = ILMHH + cdash;
    TMMHH = IMMHH + cdash;
    THMHH = IHMHH + cdash;

    TLHHH = ILHHH + cdash;
    TMHHH = IMHHH + cdash;
    THHHH = IHHHH + cdash;
! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
```

```
PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

LOOP(XVAL,1,5,0.1);

PLLLL = ILLLL*XVAL;
PMLLL = IMLLL*XVAL;
PHLLL = IHLLL*XVAL;

PLMLL = ILMLL*XVAL;
PMMLL = IMMLL*XVAL;
PHMLL = IHMLL*XVAL;

PLHLL = ILHLL*XVAL;
PMHLL = IMHLL*XVAL;
PHHLL = IHHLL*XVAL;

PLLML = ILLML*XVAL;
PMLML = IMLML*XVAL;
PHLML = IHLML*XVAL;

PLMML = ILMML*XVAL;
PMMML = IMMML*XVAL;
PHMML = IHMML*XVAL;

PLHML = ILHML*XVAL;
PMHML = IMHML*XVAL;
PHHML = IHHML*XVAL;

PLLHL = ILLHL*XVAL;
PMLHL = IMLHL*XVAL;
PHLHL = IHLHL*XVAL;

PLMHL = ILMHL*XVAL;
PMMHL = IMMHL*XVAL;
PHMHL = IHMHL*XVAL;

PLHHL = ILHHL*XVAL;
PMHHL = IMHHL*XVAL;
PHHHL = IHHHL*XVAL;

PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;
```

```
PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;

PLHLH = ILHLH*XVAL;
PMHLH = IMHLH*XVAL;
PHHLH = IHHLH*XVAL;

PLLMH = ILLMH*XVAL;
PMLMH = IMLMH*XVAL;
PHLMH = IHLMH*XVAL;

PLMMH = ILMMH*XVAL;
PMMMH = IMMMH*XVAL;
PHMMH = IHMMH*XVAL;

PLHMH = ILHMH*XVAL;
PMHMH = IMHMH*XVAL;
PHHMH = IHHMH*XVAL;
```

```
    PLLHH = ILLHH*XVAL;
    PMLHH = IMLHH*XVAL;
    PHLHH = IHLHH*XVAL;

    PLMHH = ILMHH*XVAL;
    PMMHH = IMMHH*XVAL;
    PHMHH = IHMHH*XVAL;

    PLHHH = ILHHH*XVAL;
    PMHHH = IMHHH*XVAL;
    PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 47: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path, with the other 2 moderating the Mediator-DV path with all 2-way and 3-way interactions

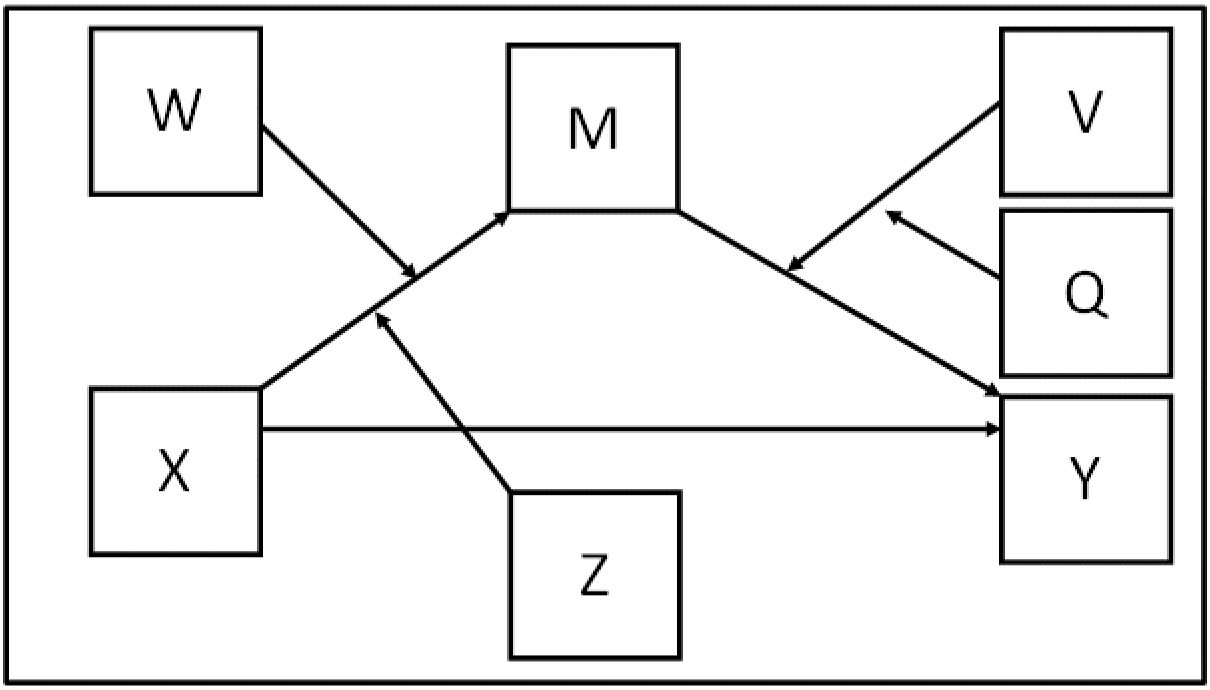
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

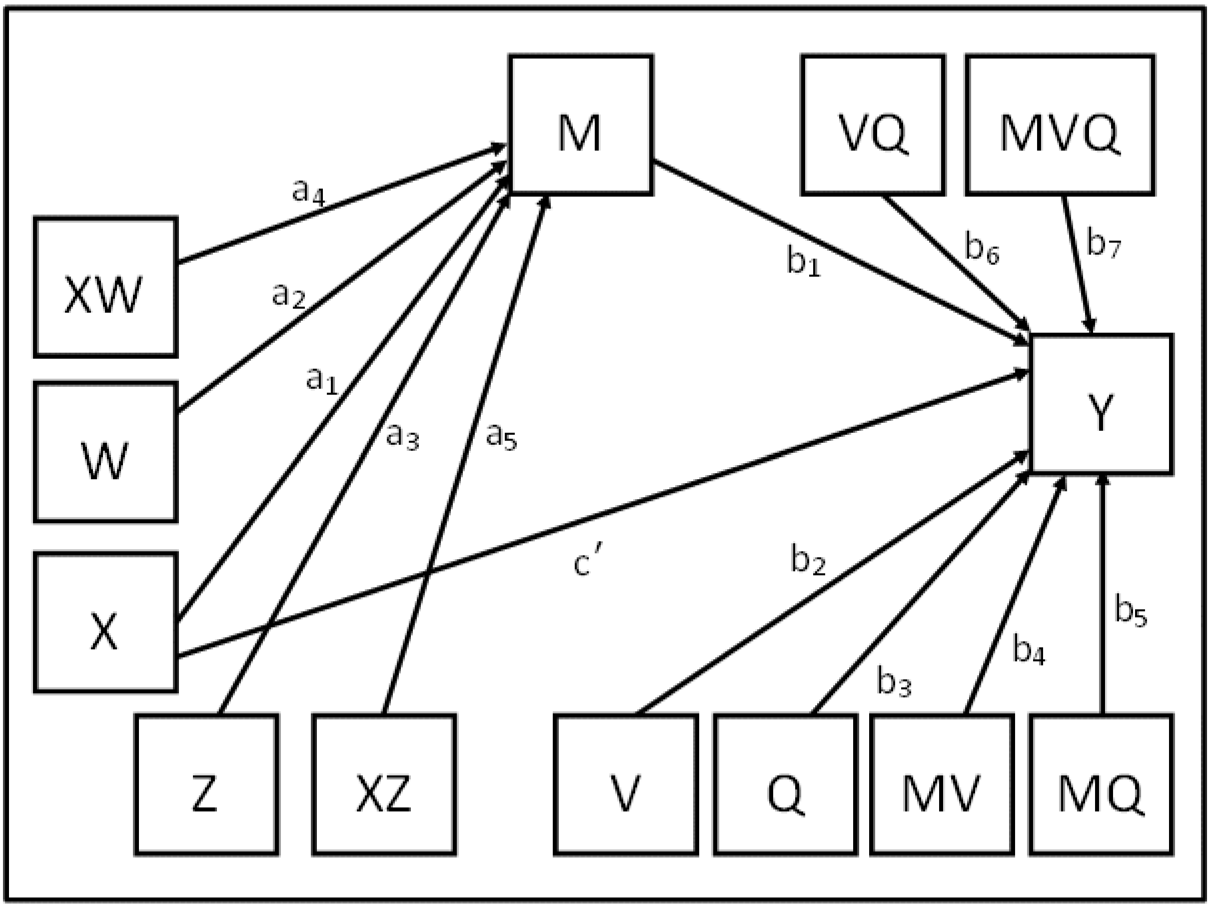
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2V + b3Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ)V + b5(a0 + a1X + a2W + a3Z + a4XW + a5XZ)Q + b6VQ + b7(a0 + a1X + a2W + a3Z + a4XW + a5XZ)VQ + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4ZV + a4b4XWV + a5b4XZV + a0b5Q + a1b5XQ + a2b5WQ + a3b5ZQ + a4b5XWQ + a5b5XZQ + b6VQ + a0b7VQ + a1b7XVQ + a2b7WVQ + a3b7ZVQ + a4b7XWVQ + a5b7XZVQ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + b2V + b3Q + a0b4V + a2b4WV + a3b4ZV + a0b5Q + a2b5WQ + a3b5ZQ + b6VQ + a0b7VQ + a2b7WVQ + a3b7ZVQ) + (a1b1 + a4b1W + a5b1Z + a1b4V + a4b4WV + a5b4ZV + a1b5Q + a4b5WQ + a5b5ZQ + a1b7VQ + a4b7WVQ + a5b7ZVQ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a1b4V + a4b4WV + a5b4ZV + a1b5Q + a4b5WQ + a5b5ZQ + a1b7VQ + a4b7WVQ + a5b7ZVQ = (a1 + a4W + a5Z) (b1 + b4V + b5Q + b7VQ)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ VQ MV MQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   VQ = V*Q;
   MVQ = M*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;
```

```
! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);
   Y ON VQ (b6);
   Y ON MVQ (b7);

   Y ON X (cdash);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
```

```
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q + a1*b7*LOW_V*LOW_Q
+
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q + a1*b7*LOW_V*LOW_Q
+
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*LOW_Q +
```

```
     a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q;
    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q + a1*b7*LOW_V*LOW_Q
+
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q + a1*b7*LOW_V*LOW_Q
+
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q;
    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q;
    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q + a1*b7*MED_V*LOW_Q
+
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q + a1*b7*MED_V*LOW_Q
+
```

```
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q + a1*b7*MED_V*LOW_Q
+
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q + a1*b7*MED_V*LOW_Q
+
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q;
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
```

```
      a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
      a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*LOW_Q +
      a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
      a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
      a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*LOW_Q +
      a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
      a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q;

    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
      a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*LOW_Q +
      a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
      a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
      a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*LOW_Q +
      a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
      a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
      a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*LOW_Q +
      a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
      a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
      a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
      a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
      a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
      a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
      a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
      a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
      a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*LOW_Q
```

```
+
     a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q + a1*b7*LOW_V*MED_Q
+
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q + a1*b7*LOW_V*MED_Q
+
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q + a1*b7*LOW_V*MED_Q
+
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q + a1*b7*LOW_V*MED_Q
+
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q;
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q;

    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
```

```
a1*b7*LOW_V*MED_Q +
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q;

    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q + a1*b7*MED_V*MED_Q
+
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q + a1*b7*MED_V*MED_Q
+
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q;
    IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q;

    ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q + a1*b7*MED_V*MED_Q
+
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q;
    IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q + a1*b7*MED_V*MED_Q
+
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q;
    IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q;
```

```
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q +
```

```
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*MED_Q
+
     a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q;

    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
```

```
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q;
    IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q;

    ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q;
    IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q;
    IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q;

    ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q;
```

```
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
```

```
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q;

    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values
```

```
TLLLL = ILLLL + cdash;
TMLLL = IMLLL + cdash;
THLLL = IHLLL + cdash;

TLMLL = ILMLL + cdash;
TMMLL = IMMLL + cdash;
THMLL = IHMLL + cdash;

TLHLL = ILHLL + cdash;
TMHLL = IMHLL + cdash;
THHLL = IHHLL + cdash;

TLLML = ILLML + cdash;
TMLML = IMLML + cdash;
THLML = IHLML + cdash;

TLMML = ILMML + cdash;
TMMML = IMMML + cdash;
THMML = IHMML + cdash;

TLHML = ILHML + cdash;
TMHML = IMHML + cdash;
THHML = IHHML + cdash;

TLLHL = ILLHL + cdash;
TMLHL = IMLHL + cdash;
THLHL = IHLHL + cdash;

TLMHL = ILMHL + cdash;
TMMHL = IMMHL + cdash;
THMHL = IHMHL + cdash;

TLHHL = ILHHL + cdash;
TMHHL = IMHHL + cdash;
THHHL = IHHHL + cdash;

TLLLM = ILLLM + cdash;
TMLLM = IMLLM + cdash;
THLLM = IHLLM + cdash;

TLMLM = ILMLM + cdash;
TMMLM = IMMLM + cdash;
THMLM = IHMLM + cdash;

TLHLM = ILHLM + cdash;
TMHLM = IMHLM + cdash;
THHLM = IHHLM + cdash;

TLLMM = ILLMM + cdash;
TMLMM = IMLMM + cdash;
THLMM = IHLMM + cdash;
```

```
TLMMM = ILMMM + cdash;
TMMMM = IMMMM + cdash;
THMMM = IHMMM + cdash;

TLHMM = ILHMM + cdash;
TMHMM = IMHMM + cdash;
THHMM = IHHMM + cdash;

TLLHM = ILLHM + cdash;
TMLHM = IMLHM + cdash;
THLHM = IHLHM + cdash;

TLMHM = ILMHM + cdash;
TMMHM = IMMHM + cdash;
THMHM = IHMHM + cdash;

TLHHM = ILHHM + cdash;
TMHHM = IMHHM + cdash;
THHHM = IHHHM + cdash;

TLLLH = ILLLH + cdash;
TMLLH = IMLLH + cdash;
THLLH = IHLLH + cdash;

TLMLH = ILMLH + cdash;
TMMLH = IMMLH + cdash;
THMLH = IHMLH + cdash;

TLHLH = ILHLH + cdash;
TMHLH = IMHLH + cdash;
THHLH = IHHLH + cdash;

TLLMH = ILLMH + cdash;
TMLMH = IMLMH + cdash;
THLMH = IHLMH + cdash;

TLMMH = ILMMH + cdash;
TMMMH = IMMMH + cdash;
THMMH = IHMMH + cdash;

TLHMH = ILHMH + cdash;
TMHMH = IMHMH + cdash;
THHMH = IHHMH + cdash;

TLLHH = ILLHH + cdash;
TMLHH = IMLHH + cdash;
THLHH = IHLHH + cdash;

TLMHH = ILMHH + cdash;
TMMHH = IMMHH + cdash;
THMHH = IHMHH + cdash;
```

```
    TLHHH = ILHHH + cdash;
    TMHHH = IMHHH + cdash;
    THHHH = IHHHH + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;

    PLMLL = ILMLL*XVAL;
    PMMLL = IMMLL*XVAL;
    PHMLL = IHMLL*XVAL;

    PLHLL = ILHLL*XVAL;
    PMHLL = IMHLL*XVAL;
    PHHLL = IHHLL*XVAL;

    PLLML = ILLML*XVAL;
    PMLML = IMLML*XVAL;
    PHLML = IHLML*XVAL;

    PLMML = ILMML*XVAL;
    PMMML = IMMML*XVAL;
    PHMML = IHMML*XVAL;

    PLHML = ILHML*XVAL;
    PMHML = IMHML*XVAL;
    PHHML = IHHML*XVAL;

    PLLHL = ILLHL*XVAL;
    PMLHL = IMLHL*XVAL;
    PHLHL = IHLHL*XVAL;
```

```
PLMHL = ILMHL*XVAL;
PMMHL = IMMHL*XVAL;
PHMHL = IHMHL*XVAL;

PLHHL = ILHHL*XVAL;
PMHHL = IMHHL*XVAL;
PHHHL = IHHHL*XVAL;

PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;
```

```
    PLHLH = ILHLH*XVAL;
    PMHLH = IMHLH*XVAL;
    PHHLH = IHHLH*XVAL;

    PLLMH = ILLMH*XVAL;
    PMLMH = IMLMH*XVAL;
    PHLMH = IHLMH*XVAL;

    PLMMH = ILMMH*XVAL;
    PMMMH = IMMMH*XVAL;
    PHMMH = IHMMH*XVAL;

    PLHMH = ILHMH*XVAL;
    PMHMH = IMHMH*XVAL;
    PHHMH = IHHMH*XVAL;

    PLLHH = ILLHH*XVAL;
    PMLHH = IMLHH*XVAL;
    PHLHH = IHLHH*XVAL;

    PLMHH = ILMHH*XVAL;
    PMMHH = IMMHH*XVAL;
    PHMHH = IHMHH*XVAL;

    PLHHH = ILHHH*XVAL;
    PMHHH = IMHHH*XVAL;
    PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 48: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path with all 2-way and 3-way interactions, with the other 2 moderating the Mediator-DV path with all 2-way and 3-way interactions

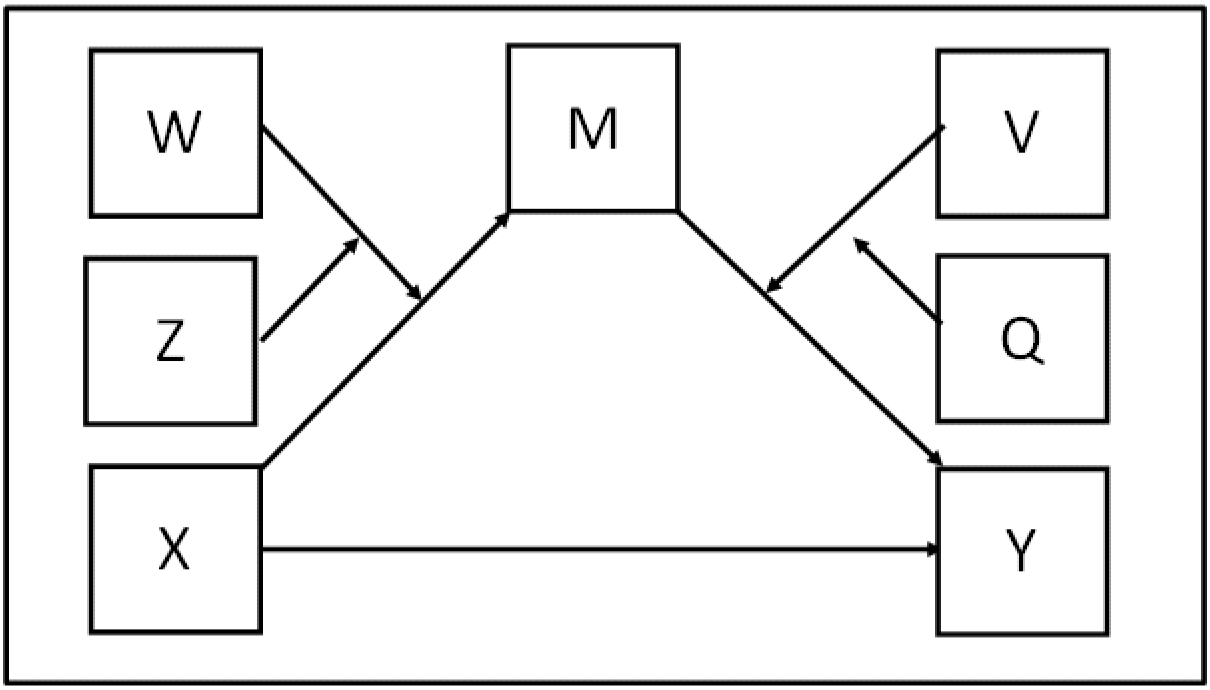
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

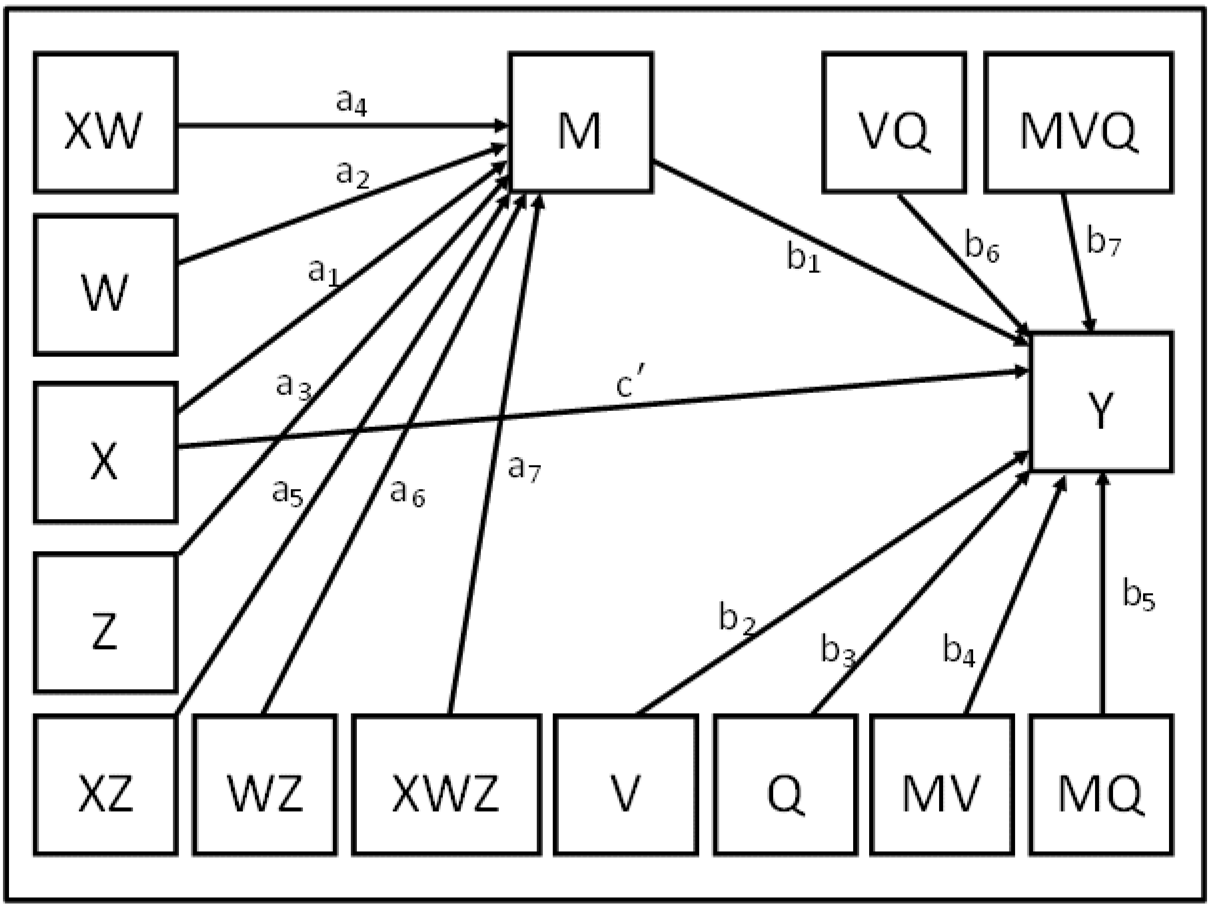
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2V + b3Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + b5(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)Q + b6VQ + b7(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)VQ + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4ZV + a4b4XWV + a5b4XZV + a6b4WZV + a7b4XWZV + a0b5Q + a1b5XQ + a2b5WQ + a3b5ZQ + a4b5XWQ + a5b5XZQ + a6b5WZQ + a7b5XWZQ + b6VQ + a0b7VQ + a1b7XVQ + a2b7WVQ + a3b7ZVQ + a4b7XWVQ + a5b7XZVQ + a6b7WZVQ + a7b7XWZVQ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + b2V + b3Q + a0b4V + a2b4WV + a3b4ZV + a6b4WZV + a0b5Q + a2b5WQ + a3b5ZQ + a6b5WZQ + b6VQ + a0b7VQ + a2b7WVQ + a3b7ZVQ + a6b7WZVQ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4V + a4b4WV + a5b4ZV + a7b4WZV + a1b5Q + a4b5WQ + a5b5ZQ + a7b5WZQ + a1b7VQ + a4b7WVQ + a5b7ZVQ + a7b7WZVQ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4V + a4b4WV + a5b4ZV + a7b4WZV + a1b5Q + a4b5WQ + a5b5ZQ + a7b5WZQ + a1b7VQ + a4b7WVQ + a5b7ZVQ + a7b7WZVQ = (a1 + a4W + a5Z + a7WZ)(b1 + b4V + b5Q + b7VQ)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ WZ VQ MV MQ XWZ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   VQ = V*Q;
   MVQ = M*V*Q;
   XWZ = X*W*Z;
```

```
ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);
   Y ON VQ (b6);
   Y ON MVQ (b7);

   Y ON X (cdash);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
```

```
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q +
```

```
a1*b7*LOW_V*LOW_Q +
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q +
     a7*b7*LOW_W*LOW_Z*LOW_V*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q +
     a7*b7*MED_W*LOW_Z*LOW_V*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q +
     a7*b7*HIGH_W*LOW_Z*LOW_V*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q +
     a7*b7*LOW_W*MED_Z*LOW_V*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q +
     a7*b7*MED_W*MED_Z*LOW_V*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
```

```
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q +
     a7*b7*HIGH_W*MED_Z*LOW_V*LOW_Q;

    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q +
     a7*b7*LOW_W*HIGH_Z*LOW_V*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q +
     a7*b7*MED_W*HIGH_Z*LOW_V*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q +
     a7*b7*HIGH_W*HIGH_Z*LOW_V*LOW_Q;

    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q +
     a7*b7*LOW_W*LOW_Z*MED_V*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q +
```

```
     a7*b7*MED_W*LOW_Z*MED_V*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q +
     a7*b7*HIGH_W*LOW_Z*MED_V*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q +
     a7*b7*LOW_W*MED_Z*MED_V*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q +
     a7*b7*MED_W*MED_Z*MED_V*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q +
     a7*b7*HIGH_W*MED_Z*MED_V*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q +
     a7*b7*LOW_W*HIGH_Z*MED_V*LOW_Q;
```

```
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q +
     a7*b7*MED_W*HIGH_Z*MED_V*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q +
     a7*b7*HIGH_W*HIGH_Z*MED_V*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q +
     a7*b7*LOW_W*LOW_Z*HIGH_V*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q +
     a7*b7*MED_W*LOW_Z*HIGH_V*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q +
     a7*b7*HIGH_W*LOW_Z*HIGH_V*LOW_Q;
```

```
    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q +
     a7*b7*LOW_W*MED_Z*HIGH_V*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q +
     a7*b7*MED_W*MED_Z*HIGH_V*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q +
     a7*b7*HIGH_W*MED_Z*HIGH_V*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q +
     a7*b7*LOW_W*HIGH_Z*HIGH_V*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q +
```

```
     a7*b7*MED_W*HIGH_Z*HIGH_V*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q +
     a7*b7*HIGH_W*HIGH_Z*HIGH_V*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q +
     a7*b7*LOW_W*LOW_Z*LOW_V*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q +
     a7*b7*MED_W*LOW_Z*LOW_V*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q +
     a7*b7*HIGH_W*LOW_Z*LOW_V*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q +
```

```
     a7*b7*LOW_W*MED_Z*LOW_V*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q +
     a7*b7*MED_W*MED_Z*LOW_V*MED_Q;
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q +
     a7*b7*HIGH_W*MED_Z*LOW_V*MED_Q;

    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q +
     a7*b7*LOW_W*HIGH_Z*LOW_V*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q +
     a7*b7*MED_W*HIGH_Z*LOW_V*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q +
     a7*b7*HIGH_W*HIGH_Z*LOW_V*MED_Q;
```

```
    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q +
     a7*b7*LOW_W*LOW_Z*MED_V*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q +
     a7*b7*MED_W*LOW_Z*MED_V*MED_Q;
    IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q +
     a7*b7*HIGH_W*LOW_Z*MED_V*MED_Q;

    ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q +
     a7*b7*LOW_W*MED_Z*MED_V*MED_Q;
    IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q +
     a7*b7*MED_W*MED_Z*MED_V*MED_Q;
    IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
```

```
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q +
     a7*b7*HIGH_W*MED_Z*MED_V*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q +
     a7*b7*LOW_W*HIGH_Z*MED_V*MED_Q;
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q +
     a7*b7*MED_W*HIGH_Z*MED_V*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q +
     a7*b7*HIGH_W*HIGH_Z*MED_V*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q +
     a7*b7*LOW_W*LOW_Z*HIGH_V*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
```

```
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q +
     a7*b7*MED_W*LOW_Z*HIGH_V*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q +
     a7*b7*HIGH_W*LOW_Z*HIGH_V*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q +
     a7*b7*LOW_W*MED_Z*HIGH_V*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q +
     a7*b7*MED_W*MED_Z*HIGH_V*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q +
     a7*b7*HIGH_W*MED_Z*HIGH_V*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
```

```
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q +
     a7*b7*LOW_W*HIGH_Z*HIGH_V*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q +
     a7*b7*MED_W*HIGH_Z*HIGH_V*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q +
     a7*b7*HIGH_W*HIGH_Z*HIGH_V*MED_Q;

    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q +
     a7*b7*LOW_W*LOW_Z*LOW_V*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q +
     a7*b7*MED_W*LOW_Z*LOW_V*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
```

```
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q +
     a7*b7*HIGH_W*LOW_Z*LOW_V*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q +
     a7*b7*LOW_W*MED_Z*LOW_V*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q +
     a7*b7*MED_W*MED_Z*LOW_V*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q +
     a7*b7*HIGH_W*MED_Z*LOW_V*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q +
     a7*b7*LOW_W*HIGH_Z*LOW_V*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
```

```
      a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
      a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
      a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
      a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q +
      a7*b7*MED_W*HIGH_Z*LOW_V*HIGH_Q;
     IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
      a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
      a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
      a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
      a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q +
      a7*b7*HIGH_W*HIGH_Z*LOW_V*HIGH_Q;

     ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
      a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
      a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
      a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
      a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q +
      a7*b7*LOW_W*LOW_Z*MED_V*HIGH_Q;
     IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
      a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
      a7*b4*MED_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
      a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
      a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q +
      a7*b7*MED_W*LOW_Z*MED_V*HIGH_Q;
     IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
      a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
      a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
      a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
      a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q +
      a7*b7*HIGH_W*LOW_Z*MED_V*HIGH_Q;

     ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
      a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
```

```
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q +
     a7*b7*LOW_W*MED_Z*MED_V*HIGH_Q;
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q +
     a7*b7*MED_W*MED_Z*MED_V*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q +
     a7*b7*HIGH_W*MED_Z*MED_V*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q +
     a7*b7*LOW_W*HIGH_Z*MED_V*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q +
     a7*b7*MED_W*HIGH_Z*MED_V*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
```

```
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q +
     a7*b7*HIGH_W*HIGH_Z*MED_V*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q +
     a7*b7*LOW_W*LOW_Z*HIGH_V*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q +
     a7*b7*MED_W*LOW_Z*HIGH_V*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q +
     a7*b7*HIGH_W*LOW_Z*HIGH_V*HIGH_Q;

    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q +
     a7*b7*LOW_W*MED_Z*HIGH_V*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
```

```
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q +
     a7*b7*MED_W*MED_Z*HIGH_V*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q +
     a7*b7*HIGH_W*MED_Z*HIGH_V*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q +
     a7*b7*LOW_W*HIGH_Z*HIGH_V*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q +
     a7*b7*MED_W*HIGH_Z*HIGH_V*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q
+
     a7*b7*HIGH_W*HIGH_Z*HIGH_V*HIGH_Q;
```

```
! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + cdash;
    TMLLL = IMLLL + cdash;
    THLLL = IHLLL + cdash;

    TLMLL = ILMLL + cdash;
    TMMLL = IMMLL + cdash;
    THMLL = IHMLL + cdash;

    TLHLL = ILHLL + cdash;
    TMHLL = IMHLL + cdash;
    THHLL = IHHLL + cdash;

    TLLML = ILLML + cdash;
    TMLML = IMLML + cdash;
    THLML = IHLML + cdash;

    TLMML = ILMML + cdash;
    TMMML = IMMML + cdash;
    THMML = IHMML + cdash;

    TLHML = ILHML + cdash;
    TMHML = IMHML + cdash;
    THHML = IHHML + cdash;

    TLLHL = ILLHL + cdash;
    TMLHL = IMLHL + cdash;
    THLHL = IHLHL + cdash;

    TLMHL = ILMHL + cdash;
    TMMHL = IMMHL + cdash;
    THMHL = IHMHL + cdash;

    TLHHL = ILHHL + cdash;
    TMHHL = IMHHL + cdash;
    THHHL = IHHHL + cdash;

    TLLLM = ILLLM + cdash;
    TMLLM = IMLLM + cdash;
    THLLM = IHLLM + cdash;

    TLMLM = ILMLM + cdash;
    TMMLM = IMMLM + cdash;
    THMLM = IHMLM + cdash;

    TLHLM = ILHLM + cdash;
    TMHLM = IMHLM + cdash;
    THHLM = IHHLM + cdash;
```

```
TLLMM = ILLMM + cdash;
TMLMM = IMLMM + cdash;
THLMM = IHLMM + cdash;

TLMMM = ILMMM + cdash;
TMMMM = IMMMM + cdash;
THMMM = IHMMM + cdash;

TLHMM = ILHMM + cdash;
TMHMM = IMHMM + cdash;
THHMM = IHHMM + cdash;

TLLHM = ILLHM + cdash;
TMLHM = IMLHM + cdash;
THLHM = IHLHM + cdash;

TLMHM = ILMHM + cdash;
TMMHM = IMMHM + cdash;
THMHM = IHMHM + cdash;

TLHHM = ILHHM + cdash;
TMHHM = IMHHM + cdash;
THHHM = IHHHM + cdash;

TLLLH = ILLLH + cdash;
TMLLH = IMLLH + cdash;
THLLH = IHLLH + cdash;

TLMLH = ILMLH + cdash;
TMMLH = IMMLH + cdash;
THMLH = IHMLH + cdash;

TLHLH = ILHLH + cdash;
TMHLH = IMHLH + cdash;
THHLH = IHHLH + cdash;

TLLMH = ILLMH + cdash;
TMLMH = IMLMH + cdash;
THLMH = IHLMH + cdash;

TLMMH = ILMMH + cdash;
TMMMH = IMMMH + cdash;
THMMH = IHMMH + cdash;

TLHMH = ILHMH + cdash;
TMHMH = IMHMH + cdash;
THHMH = IHHMH + cdash;

TLLHH = ILLHH + cdash;
TMLHH = IMLHH + cdash;
THLHH = IHLHH + cdash;
```

```
    TLMHH = ILMHH + cdash;
    TMMHH = IMMHH + cdash;
    THMHH = IHMHH + cdash;

    TLHHH = ILHHH + cdash;
    TMHHH = IMHHH + cdash;
    THHHH = IHHHH + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;

    PLMLL = ILMLL*XVAL;
    PMMLL = IMMLL*XVAL;
    PHMLL = IHMLL*XVAL;

    PLHLL = ILHLL*XVAL;
    PMHLL = IMHLL*XVAL;
    PHHLL = IHHLL*XVAL;

    PLLML = ILLML*XVAL;
    PMLML = IMLML*XVAL;
    PHLML = IHLML*XVAL;

    PLMML = ILMML*XVAL;
    PMMML = IMMML*XVAL;
    PHMML = IHMML*XVAL;

    PLHML = ILHML*XVAL;
    PMHML = IMHML*XVAL;
    PHHML = IHHML*XVAL;
```

```
PLLHL = ILLHL*XVAL;
PMLHL = IMLHL*XVAL;
PHLHL = IHLHL*XVAL;

PLMHL = ILMHL*XVAL;
PMMHL = IMMHL*XVAL;
PHMHL = IHMHL*XVAL;

PLHHL = ILHHL*XVAL;
PMHHL = IMHHL*XVAL;
PHHHL = IHHHL*XVAL;

PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;
```

```
    PLMLH = ILMLH*XVAL;
    PMMLH = IMMLH*XVAL;
    PHMLH = IHMLH*XVAL;

    PLHLH = ILHLH*XVAL;
    PMHLH = IMHLH*XVAL;
    PHHLH = IHHLH*XVAL;

    PLLMH = ILLMH*XVAL;
    PMLMH = IMLMH*XVAL;
    PHLMH = IHLMH*XVAL;

    PLMMH = ILMMH*XVAL;
    PMMMH = IMMMH*XVAL;
    PHMMH = IHMMH*XVAL;

    PLHMH = ILHMH*XVAL;
    PMHMH = IMHMH*XVAL;
    PHHMH = IHHMH*XVAL;

    PLLHH = ILLHH*XVAL;
    PMLHH = IMLHH*XVAL;
    PHLHH = IHLHH*XVAL;

    PLMHH = ILMHH*XVAL;
    PMMHH = IMMHH*XVAL;
    PHMHH = IHMHH*XVAL;

    PLHHH = ILHHH*XVAL;
    PMHHH = IMHHH*XVAL;
    PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 49: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path, with the other 2 moderating the Mediator-DV path

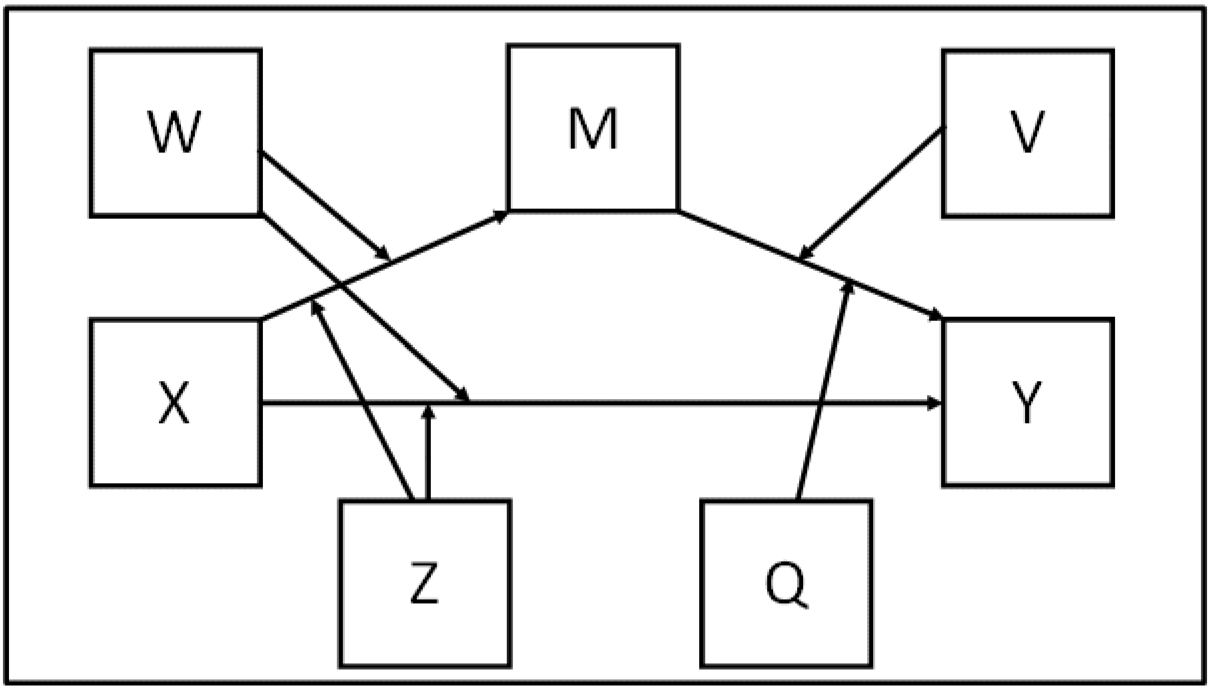
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

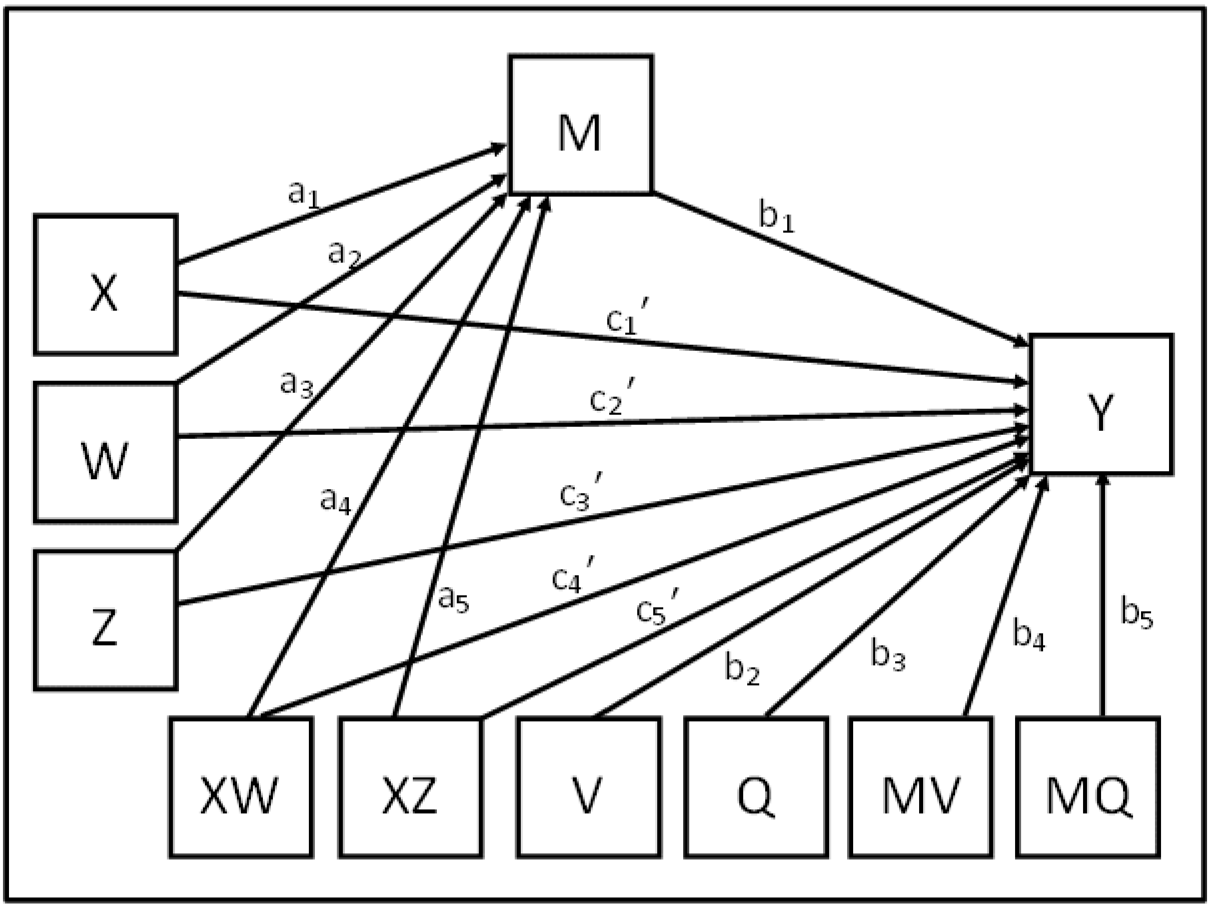
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2V + b3Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ)V + b5(a0 + a1X + a2W + a3Z + a4XW + a5XZ)Q + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4ZV + a4b4XWV + a5b4XZV + a0b5Q + a1b5XQ

+ a2b5WQ + a3b5ZQ + a4b5XWQ + a5b5XZQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + b2V + b3Q + a0b4V + a2b4WV + a3b4ZV + a0b5Q + a2b5WQ + a3b5ZQ + c2'W + c3'Z) + (a1b1 + a4b1W + a5b1Z + a1b4V + a4b4WV + a5b4ZV + a1b5Q + a4b5WQ + a5b5ZQ + c1' + c4'W + c5'Z)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a1b4V + a4b4WV + a5b4ZV + a1b5Q + a4b5WQ + a5b5ZQ = (a1 + a4W + a5Z)(b1 + b4V + b5Q)

One direct effect of X on Y, conditional on W, Z: c1' + c4'W + c5'Z

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ MV MQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
```

```
    Y ON MV (b4);
    Y ON MQ (b5);

    Y ON X (cdash1);
    Y ON W (cdash2);
    Y ON Z (cdash3);
    Y ON XW (cdash4);
    Y ON XZ (cdash5);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
```

```
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*LOW_Q +
```

```
 a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q;

ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
 a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
 a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;

ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;

ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q;

ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
 a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
 a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
```

```
IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q;
ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*LOW_Q +
 a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q;
ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
 a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
 a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*LOW_Q
+
 a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q;
ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_V +
 a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q;
IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*LOW_V +
```

```
 a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q;
IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*MED_Q +
 a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q;

ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_V +
 a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q;
IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*LOW_V +
 a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q;
IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*MED_Q +
 a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q;

ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
 a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
 a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*MED_Q +
 a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q;

ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q;
IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q;
IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*MED_Q +
 a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q;

ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*MED_Q +
 a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q;
IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q;
IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*MED_V
```

```
+
     a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
```

```
 a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*MED_Q +
 a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q;
IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*MED_Q
+
 a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q;

ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_V +
 a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*LOW_V +
 a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;

ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_V +
 a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*LOW_V +
 a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;

ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
 a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
 a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
 a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;

ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
```

```
IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;

ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*MED_V +
 a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_V +
 a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;

ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
 a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
 a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
 a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*HIGH_Q +
 a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;

ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
 a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
 a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;
IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
 a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*HIGH_Q
+
 a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q;

ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
 a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*HIGH_Q +
 a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
```

```
     a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + DLOW_LOZ;
    TMLLL = IMLLL + DMEW_LOZ;
    THLLL = IHLLL + DHIW_LOZ;

    TLMLL = ILMLL + DLOW_MEZ;
    TMMLL = IMMLL + DMEW_MEZ;
    THMLL = IHMLL + DHIW_MEZ;
```

```
TLHLL = ILHLL + DLOW_HIZ;
TMHLL = IMHLL + DMEW_HIZ;
THHLL = IHHLL + DHIW_HIZ;

TLLML = ILLML + DLOW_LOZ;
TMLML = IMLML + DMEW_LOZ;
THLML = IHLML + DHIW_LOZ;

TLMML = ILMML + DLOW_MEZ;
TMMML = IMMML + DMEW_MEZ;
THMML = IHMML + DHIW_MEZ;

TLHML = ILHML + DLOW_HIZ;
TMHML = IMHML + DMEW_HIZ;
THHML = IHHML + DHIW_HIZ;

TLLHL = ILLHL + DLOW_LOZ;
TMLHL = IMLHL + DMEW_LOZ;
THLHL = IHLHL + DHIW_LOZ;

TLMHL = ILMHL + DLOW_MEZ;
TMMHL = IMMHL + DMEW_MEZ;
THMHL = IHMHL + DHIW_MEZ;

TLHHL = ILHHL + DLOW_HIZ;
TMHHL = IMHHL + DMEW_HIZ;
THHHL = IHHHL + DHIW_HIZ;

TLLLM = ILLLM + DLOW_LOZ;
TMLLM = IMLLM + DMEW_LOZ;
THLLM = IHLLM + DHIW_LOZ;

TLMLM = ILMLM + DLOW_MEZ;
TMMLM = IMMLM + DMEW_MEZ;
THMLM = IHMLM + DHIW_MEZ;

TLHLM = ILHLM + DLOW_HIZ;
TMHLM = IMHLM + DMEW_HIZ;
THHLM = IHHLM + DHIW_HIZ;

TLLMM = ILLMM + DLOW_LOZ;
TMLMM = IMLMM + DMEW_LOZ;
THLMM = IHLMM + DHIW_LOZ;

TLMMM = ILMMM + DLOW_MEZ;
TMMMM = IMMMM + DMEW_MEZ;
THMMM = IHMMM + DHIW_MEZ;

TLHMM = ILHMM + DLOW_HIZ;
TMHMM = IMHMM + DMEW_HIZ;
THHMM = IHHMM + DHIW_HIZ;
```

```
    TLLHM = ILLHM + DLOW_LOZ;
    TMLHM = IMLHM + DMEW_LOZ;
    THLHM = IHLHM + DHIW_LOZ;

    TLMHM = ILMHM + DLOW_MEZ;
    TMMHM = IMMHM + DMEW_MEZ;
    THMHM = IHMHM + DHIW_MEZ;

    TLHHM = ILHHM + DLOW_HIZ;
    TMHHM = IMHHM + DMEW_HIZ;
    THHHM = IHHHM + DHIW_HIZ;

    TLLLH = ILLLH + DLOW_LOZ;
    TMLLH = IMLLH + DMEW_LOZ;
    THLLH = IHLLH + DHIW_LOZ;

    TLMLH = ILMLH + DLOW_MEZ;
    TMMLH = IMMLH + DMEW_MEZ;
    THMLH = IHMLH + DHIW_MEZ;

    TLHLH = ILHLH + DLOW_HIZ;
    TMHLH = IMHLH + DMEW_HIZ;
    THHLH = IHHLH + DHIW_HIZ;

    TLLMH = ILLMH + DLOW_LOZ;
    TMLMH = IMLMH + DMEW_LOZ;
    THLMH = IHLMH + DHIW_LOZ;

    TLMMH = ILMMH + DLOW_MEZ;
    TMMMH = IMMMH + DMEW_MEZ;
    THMMH = IHMMH + DHIW_MEZ;

    TLHMH = ILHMH + DLOW_HIZ;
    TMHMH = IMHMH + DMEW_HIZ;
    THHMH = IHHMH + DHIW_HIZ;

    TLLHH = ILLHH + DLOW_LOZ;
    TMLHH = IMLHH + DMEW_LOZ;
    THLHH = IHLHH + DHIW_LOZ;

    TLMHH = ILMHH + DLOW_MEZ;
    TMMHH = IMMHH + DMEW_MEZ;
    THMHH = IHMHH + DHIW_MEZ;

    TLHHH = ILHHH + DLOW_HIZ;
    TMHHH = IMHHH + DMEW_HIZ;
    THHHH = IHHHH + DHIW_HIZ;
! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
```

```
by
! logical min and max limits of predictor X used in analysis
    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;

    PLMLL = ILMLL*XVAL;
    PMMLL = IMMLL*XVAL;
    PHMLL = IHMLL*XVAL;

    PLHLL = ILHLL*XVAL;
    PMHLL = IMHLL*XVAL;
    PHHLL = IHHLL*XVAL;

    PLLML = ILLML*XVAL;
    PMLML = IMLML*XVAL;
    PHLML = IHLML*XVAL;

    PLMML = ILMML*XVAL;
    PMMML = IMMML*XVAL;
    PHMML = IHMML*XVAL;

    PLHML = ILHML*XVAL;
    PMHML = IMHML*XVAL;
    PHHML = IHHML*XVAL;

    PLLHL = ILLHL*XVAL;
    PMLHL = IMLHL*XVAL;
    PHLHL = IHLHL*XVAL;

    PLMHL = ILMHL*XVAL;
    PMMHL = IMMHL*XVAL;
    PHMHL = IHMHL*XVAL;

    PLHHL = ILHHL*XVAL;
    PMHHL = IMHHL*XVAL;
    PHHHL = IHHHL*XVAL;
```

```
PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;

PLHLH = ILHLH*XVAL;
PMHLH = IMHLH*XVAL;
PHHLH = IHHLH*XVAL;

PLLMH = ILLMH*XVAL;
PMLMH = IMLMH*XVAL;
PHLMH = IHLMH*XVAL;
```

```
    PLMMH = ILMMH*XVAL;
    PMMMH = IMMMH*XVAL;
    PHMMH = IHMMH*XVAL;

    PLHMH = ILHMH*XVAL;
    PMHMH = IMHMH*XVAL;
    PHHMH = IHHMH*XVAL;

    PLLHH = ILLHH*XVAL;
    PMLHH = IMLHH*XVAL;
    PHLHH = IHLHH*XVAL;

    PLMHH = ILMHH*XVAL;
    PMMHH = IMMHH*XVAL;
    PHMHH = IHMHH*XVAL;

    PLHHH = ILHHH*XVAL;
    PMHHH = IMHHH*XVAL;
    PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 50: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path

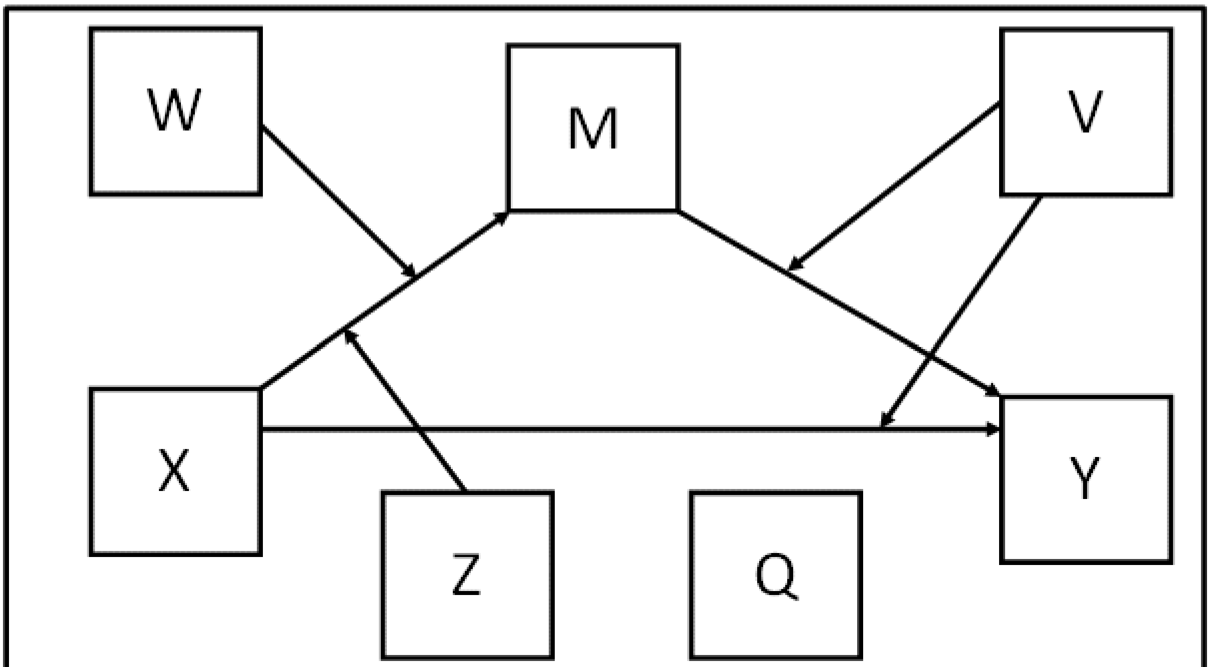
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

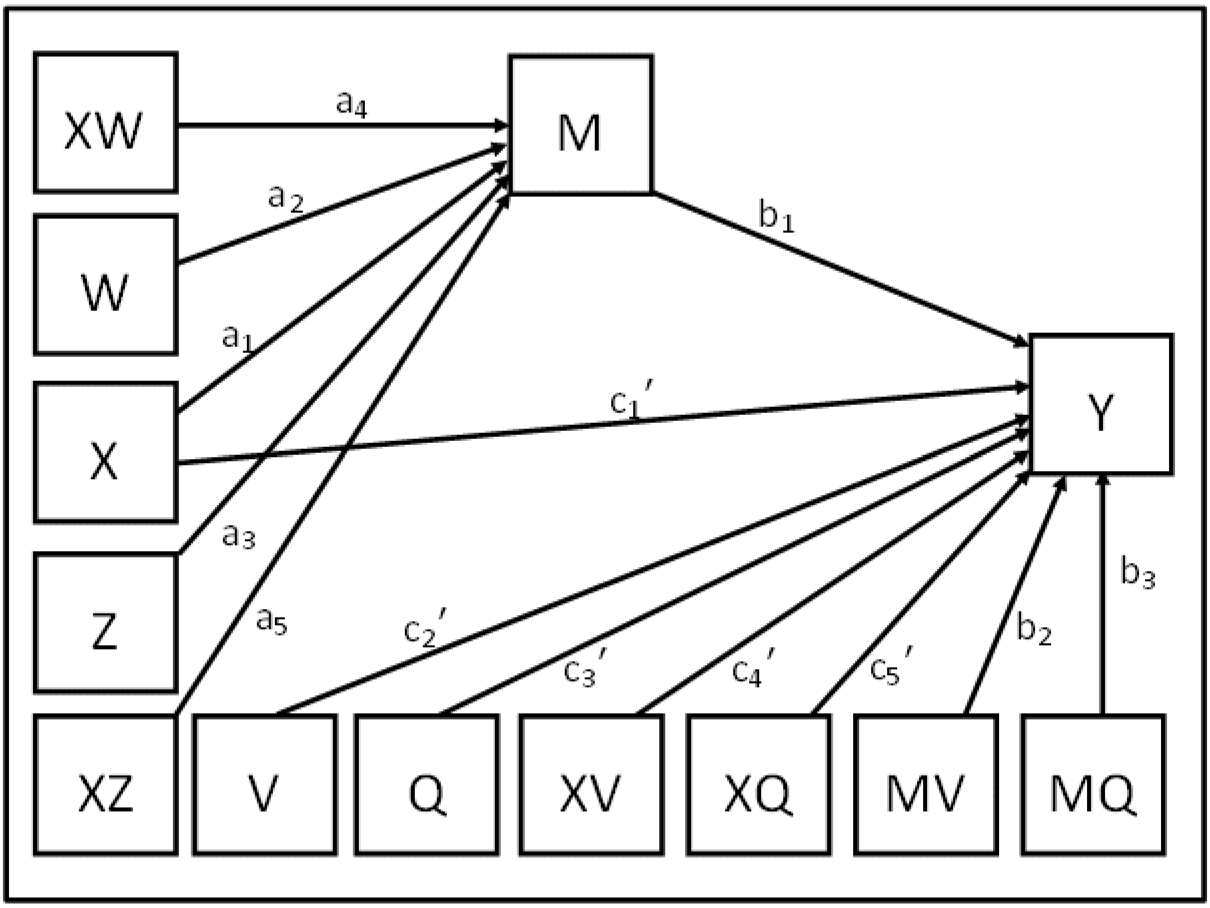
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ)V + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ)Q + c1'X + c2'V + c3'Q + c4'XV + c5'XQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a0b2V + a1b2XV + a2b2WV + a3b2ZV + a4b2XWV + a5b2XZV + a0b3Q + a1b3XQ + a2b3WQ + a3b3ZQ + a4b3XWQ + a5b3XZQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a0b2V + a2b2WV + a3b2ZV + a0b3Q + a2b3WQ + a3b3ZQ + c2'V + c3'Q) + (a1b1 + a4b1W + a5b1Z + a1b2V + a4b2WV + a5b2ZV + a1b3Q + a4b3WQ + a5b3ZQ + c1' + c4'V + c5'Q)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a1b2V + a4b2WV + a5b2ZV + a1b3Q + a4b3WQ + a5b3ZQ = (a1 + a4W + a5Z)(b1 + b2V + b3Q)

One direct effect of X on Y, conditional on V, Q:

c1' + c4'V + c5'Q

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ XV XQ MV MQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   XQ = X*Q;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON MV (b2);
    Y ON MQ (b3);

    Y ON X (cdash1);
    Y ON V (cdash2);
    Y ON Q (cdash3);
    Y ON XV (cdash4);
    Y ON XQ (cdash5);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
     NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
     ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
     ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
     ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
     ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
     ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
     ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
     ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
     ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
     ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
     DLOV_LOQ DMEV_LOQ DHIV_LOQ DLOV_MEQ DMEV_MEQ DHIV_MEQ
     DLOV_HIQ DMEV_HIQ DHIV_HIQ
     TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
     TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
     TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
     TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
```

```
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_V +
     a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*LOW_Z*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*LOW_V +
     a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*LOW_Z*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*LOW_Z*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_V +
     a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*LOW_Q +
```

```
 a4*b3*LOW_W*LOW_Q + a5*b3*MED_Z*LOW_Q;
IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*LOW_V +
 a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*LOW_Q +
 a4*b3*MED_W*LOW_Q + a5*b3*MED_Z*LOW_Q;
IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*LOW_V
+
 a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*LOW_Q +
 a4*b3*HIGH_W*LOW_Q + a5*b3*MED_Z*LOW_Q;
ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
 a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*LOW_Q +
 a4*b3*LOW_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q;
IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
 a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*LOW_Q +
 a4*b3*MED_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q;
IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
 a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*LOW_Q +
 a4*b3*HIGH_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q;
ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*MED_V +
 a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*LOW_Q +
 a4*b3*LOW_W*LOW_Q + a5*b3*LOW_Z*LOW_Q;
IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_V +
 a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*LOW_Q +
 a4*b3*MED_W*LOW_Q + a5*b3*LOW_Z*LOW_Q;
IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*MED_V
+
 a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*LOW_Q +
 a4*b3*HIGH_W*LOW_Q + a5*b3*LOW_Z*LOW_Q;
ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*MED_V +
 a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*LOW_Q +
 a4*b3*LOW_W*LOW_Q + a5*b3*MED_Z*LOW_Q;
IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_V +
 a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*LOW_Q +
 a4*b3*MED_W*LOW_Q + a5*b3*MED_Z*LOW_Q;
IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*MED_V
+
 a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*LOW_Q +
 a4*b3*HIGH_W*LOW_Q + a5*b3*MED_Z*LOW_Q;
ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
 a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*LOW_Q +
 a4*b3*LOW_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q;
IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*MED_V
```

```
+
     a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q;
    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*LOW_Z*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*LOW_Z*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*LOW_Z*LOW_Q;
    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*MED_Z*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*MED_Z*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*MED_Z*LOW_Q;
    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*LOW_Q
+
     a4*b3*HIGH_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q;
```

```
ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_V +
 a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*MED_Q +
 a4*b3*LOW_W*MED_Q + a5*b3*LOW_Z*MED_Q;
IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*LOW_V +
 a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*MED_Q +
 a4*b3*MED_W*MED_Q + a5*b3*LOW_Z*MED_Q;
IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
 a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*MED_Q +
 a4*b3*HIGH_W*MED_Q + a5*b3*LOW_Z*MED_Q;

ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_V +
 a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*MED_Q +
 a4*b3*LOW_W*MED_Q + a5*b3*MED_Z*MED_Q;
IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*LOW_V +
 a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*MED_Q +
 a4*b3*MED_W*MED_Q + a5*b3*MED_Z*MED_Q;
IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*LOW_V
+
 a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*MED_Q +
 a4*b3*HIGH_W*MED_Q + a5*b3*MED_Z*MED_Q;

ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
 a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*MED_Q +
 a4*b3*LOW_W*MED_Q + a5*b3*HIGH_Z*MED_Q;
IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
 a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*MED_Q +
 a4*b3*MED_W*MED_Q + a5*b3*HIGH_Z*MED_Q;
IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
 a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*MED_Q +
 a4*b3*HIGH_W*MED_Q + a5*b3*HIGH_Z*MED_Q;

ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*MED_V +
 a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*MED_Q +
 a4*b3*LOW_W*MED_Q + a5*b3*LOW_Z*MED_Q;
IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_V +
 a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*MED_Q +
 a4*b3*MED_W*MED_Q + a5*b3*LOW_Z*MED_Q;
IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*MED_V
+
 a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*MED_Q +
 a4*b3*HIGH_W*MED_Q + a5*b3*LOW_Z*MED_Q;

ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*MED_V +
 a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*MED_Q +
 a4*b3*LOW_W*MED_Q + a5*b3*MED_Z*MED_Q;
```

```
IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_V +
 a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*MED_Q +
 a4*b3*MED_W*MED_Q + a5*b3*MED_Z*MED_Q;
IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*MED_V
+
 a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*MED_Q +
 a4*b3*HIGH_W*MED_Q + a5*b3*MED_Z*MED_Q;
ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
 a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*MED_Q +
 a4*b3*LOW_W*MED_Q + a5*b3*HIGH_Z*MED_Q;
IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
 a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*MED_Q +
 a4*b3*MED_W*MED_Q + a5*b3*HIGH_Z*MED_Q;
IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
 a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*MED_Q +
 a4*b3*HIGH_W*MED_Q + a5*b3*HIGH_Z*MED_Q;
ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
 a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*MED_Q +
 a4*b3*LOW_W*MED_Q + a5*b3*LOW_Z*MED_Q;
IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
 a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*MED_Q +
 a4*b3*MED_W*MED_Q + a5*b3*LOW_Z*MED_Q;
IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
 a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*MED_Q +
 a4*b3*HIGH_W*MED_Q + a5*b3*LOW_Z*MED_Q;
ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
 a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*MED_Q +
 a4*b3*LOW_W*MED_Q + a5*b3*MED_Z*MED_Q;
IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
 a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*MED_Q +
 a4*b3*MED_W*MED_Q + a5*b3*MED_Z*MED_Q;
IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
 a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*MED_Q +
 a4*b3*HIGH_W*MED_Q + a5*b3*MED_Z*MED_Q;
ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
```

```
 a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*MED_Q +
 a4*b3*LOW_W*MED_Q + a5*b3*HIGH_Z*MED_Q;
IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
 a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*MED_Q +
 a4*b3*MED_W*MED_Q + a5*b3*HIGH_Z*MED_Q;
IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
 a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*MED_Q
+
 a4*b3*HIGH_W*MED_Q + a5*b3*HIGH_Z*MED_Q;

ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_V +
 a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*HIGH_Q +
 a4*b3*LOW_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q;
IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*LOW_V +
 a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*HIGH_Q +
 a4*b3*MED_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q;
IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
 a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*HIGH_Q +
 a4*b3*HIGH_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q;

ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_V +
 a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*HIGH_Q +
 a4*b3*LOW_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q;
IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*LOW_V +
 a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*HIGH_Q +
 a4*b3*MED_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q;
IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*LOW_V
+
 a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*HIGH_Q +
 a4*b3*HIGH_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q;

ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
 a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
 a4*b3*LOW_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q;
IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
 a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
 a4*b3*MED_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q;
IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
 a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
 a4*b3*HIGH_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q;

ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*MED_V +
 a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*HIGH_Q +
```

```
 a4*b3*LOW_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q;
IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_V +
 a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*HIGH_Q +
 a4*b3*MED_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q;
IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*MED_V
+
 a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*HIGH_Q +
 a4*b3*HIGH_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q;

ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*MED_V +
 a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*HIGH_Q +
 a4*b3*LOW_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q;
IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_V +
 a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*HIGH_Q +
 a4*b3*MED_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q;
IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*MED_V
+
 a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*HIGH_Q +
 a4*b3*HIGH_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q;

ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
 a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*HIGH_Q +
 a4*b3*LOW_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q;
IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
 a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*HIGH_Q +
 a4*b3*MED_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q;
IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
 a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*HIGH_Q +
 a4*b3*HIGH_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q;

ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
 a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
 a4*b3*LOW_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q;
IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
 a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
 a4*b3*MED_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q;
IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
 a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*HIGH_Q
+
 a4*b3*HIGH_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q;

ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
```

```
     a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*HIGH_Q +
     a4*b3*LOW_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*HIGH_Q +
     a4*b3*MED_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*HIGH_Q
+
     a4*b3*HIGH_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*HIGH_Q
+
     a4*b3*LOW_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*HIGH_Q
+
     a4*b3*MED_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*HIGH_Q
+
     a4*b3*HIGH_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOV_LOQ = cdash1 + cdash4*LOW_V + cdash5*LOW_Q;
    DMEV_LOQ = cdash1 + cdash4*MED_W + cdash5*LOW_Q;
    DHIV_LOQ = cdash1 + cdash4*HIGH_V + cdash5*LOW_Q;

    DLOV_MEQ = cdash1 + cdash4*LOW_V + cdash5*MED_Q;
    DMEV_MEQ = cdash1 + cdash4*MED_W + cdash5*MED_Q;
    DHIV_MEQ = cdash1 + cdash4*HIGH_V + cdash5*MED_Q;

    DLOV_HIQ = cdash1 + cdash4*LOW_V + cdash5*HIGH_Q;
    DMEV_HIQ = cdash1 + cdash4*MED_W + cdash5*HIGH_Q;
    DHIV_HIQ = cdash1 + cdash4*HIGH_V + cdash5*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + DLOV_LOQ;
    TMLLL = IMLLL + DLOV_LOQ;
    THLLL = IHLLL + DLOV_LOQ;
```

```
TLMLL = ILMLL + DLOV_LOQ;
TMMLL = IMMLL + DLOV_LOQ;
THMLL = IHMLL + DLOV_LOQ;

TLHLL = ILHLL + DLOV_LOQ;
TMHLL = IMHLL + DLOV_LOQ;
THHLL = IHHLL + DLOV_LOQ;

TLLML = ILLML + DMEV_LOQ;
TMLML = IMLML + DMEV_LOQ;
THLML = IHLML + DMEV_LOQ;

TLMML = ILMML + DMEV_LOQ;
TMMML = IMMML + DMEV_LOQ;
THMML = IHMML + DMEV_LOQ;

TLHML = ILHML + DMEV_LOQ;
TMHML = IMHML + DMEV_LOQ;
THHML = IHHML + DMEV_LOQ;

TLLHL = ILLHL + DHIV_LOQ;
TMLHL = IMLHL + DHIV_LOQ;
THLHL = IHLHL + DHIV_LOQ;

TLMHL = ILMHL + DHIV_LOQ;
TMMHL = IMMHL + DHIV_LOQ;
THMHL = IHMHL + DHIV_LOQ;

TLHHL = ILHHL + DHIV_LOQ;
TMHHL = IMHHL + DHIV_LOQ;
THHHL = IHHHL + DHIV_LOQ;

TLLLM = ILLLM + DLOV_MEQ;
TMLLM = IMLLM + DLOV_MEQ;
THLLM = IHLLM + DLOV_MEQ;

TLMLM = ILMLM + DLOV_MEQ;
TMMLM = IMMLM + DLOV_MEQ;
THMLM = IHMLM + DLOV_MEQ;

TLHLM = ILHLM + DLOV_MEQ;
TMHLM = IMHLM + DLOV_MEQ;
THHLM = IHHLM + DLOV_MEQ;

TLLMM = ILLMM + DMEV_MEQ;
TMLMM = IMLMM + DMEV_MEQ;
THLMM = IHLMM + DMEV_MEQ;

TLMMM = ILMMM + DMEV_MEQ;
TMMMM = IMMMM + DMEV_MEQ;
THMMM = IHMMM + DMEV_MEQ;
```

```
TLHMM = ILHMM + DMEV_MEQ;
TMHMM = IMHMM + DMEV_MEQ;
THHMM = IHHMM + DMEV_MEQ;

TLLHM = ILLHM + DHIV_MEQ;
TMLHM = IMLHM + DHIV_MEQ;
THLHM = IHLHM + DHIV_MEQ;

TLMHM = ILMHM + DHIV_MEQ;
TMMHM = IMMHM + DHIV_MEQ;
THMHM = IHMHM + DHIV_MEQ;

TLHHM = ILHHM + DHIV_MEQ;
TMHHM = IMHHM + DHIV_MEQ;
THHHM = IHHHM + DHIV_MEQ;

TLLLH = ILLLH + DLOV_HIQ;
TMLLH = IMLLH + DLOV_HIQ;
THLLH = IHLLH + DLOV_HIQ;

TLMLH = ILMLH + DLOV_HIQ;
TMMLH = IMMLH + DLOV_HIQ;
THMLH = IHMLH + DLOV_HIQ;

TLHLH = ILHLH + DLOV_HIQ;
TMHLH = IMHLH + DLOV_HIQ;
THHLH = IHHLH + DLOV_HIQ;

TLLMH = ILLMH + DMEV_HIQ;
TMLMH = IMLMH + DMEV_HIQ;
THLMH = IHLMH + DMEV_HIQ;

TLMMH = ILMMH + DMEV_HIQ;
TMMMH = IMMMH + DMEV_HIQ;
THMMH = IHMMH + DMEV_HIQ;

TLHMH = ILHMH + DMEV_HIQ;
TMHMH = IMHMH + DMEV_HIQ;
THHMH = IHHMH + DMEV_HIQ;

TLLHH = ILLHH + DHIV_HIQ;
TMLHH = IMLHH + DHIV_HIQ;
THLHH = IHLHH + DHIV_HIQ;

TLMHH = ILMHH + DHIV_HIQ;
TMMHH = IMMHH + DHIV_HIQ;
THMHH = IHMHH + DHIV_HIQ;

TLHHH = ILHHH + DHIV_HIQ;
TMHHH = IMHHH + DHIV_HIQ;
THHHH = IHHHH + DHIV_HIQ;
```

```
! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;

    PLMLL = ILMLL*XVAL;
    PMMLL = IMMLL*XVAL;
    PHMLL = IHMLL*XVAL;

    PLHLL = ILHLL*XVAL;
    PMHLL = IMHLL*XVAL;
    PHHLL = IHHLL*XVAL;

    PLLML = ILLML*XVAL;
    PMLML = IMLML*XVAL;
    PHLML = IHLML*XVAL;

    PLMML = ILMML*XVAL;
    PMMML = IMMML*XVAL;
    PHMML = IHMML*XVAL;

    PLHML = ILHML*XVAL;
    PMHML = IMHML*XVAL;
    PHHML = IHHML*XVAL;

    PLLHL = ILLHL*XVAL;
    PMLHL = IMLHL*XVAL;
    PHLHL = IHLHL*XVAL;

    PLMHL = ILMHL*XVAL;
    PMMHL = IMMHL*XVAL;
    PHMHL = IHMHL*XVAL;
```

```
PLHHL = ILHHL*XVAL;
PMHHL = IMHHL*XVAL;
PHHHL = IHHHL*XVAL;

PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;

PLHLH = ILHLH*XVAL;
PMHLH = IMHLH*XVAL;
PHHLH = IHHLH*XVAL;
```

```
    PLLMH = ILLMH*XVAL;
    PMLMH = IMLMH*XVAL;
    PHLMH = IHLMH*XVAL;

    PLMMH = ILMMH*XVAL;
    PMMMH = IMMMH*XVAL;
    PHMMH = IHMMH*XVAL;

    PLHMH = ILHMH*XVAL;
    PMHMH = IMHMH*XVAL;
    PHHMH = IHHMH*XVAL;

    PLLHH = ILLHH*XVAL;
    PMLHH = IMLHH*XVAL;
    PHLHH = IHLHH*XVAL;

    PLMHH = ILMHH*XVAL;
    PMMHH = IMMHH*XVAL;
    PHMHH = IHMHH*XVAL;

    PLHHH = ILHHH*XVAL;
    PMHHH = IMHHH*XVAL;
    PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 51: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path with all 2-way and 3-way interactions, with the other 2 moderating the Mediator-DV path

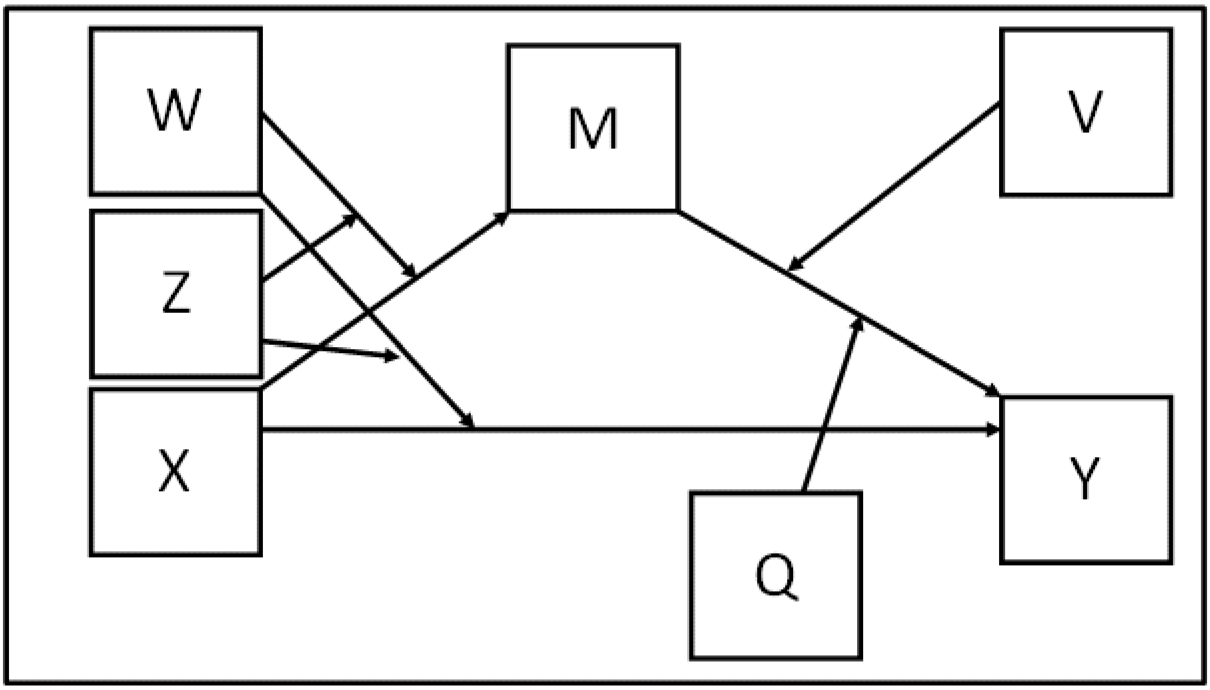
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

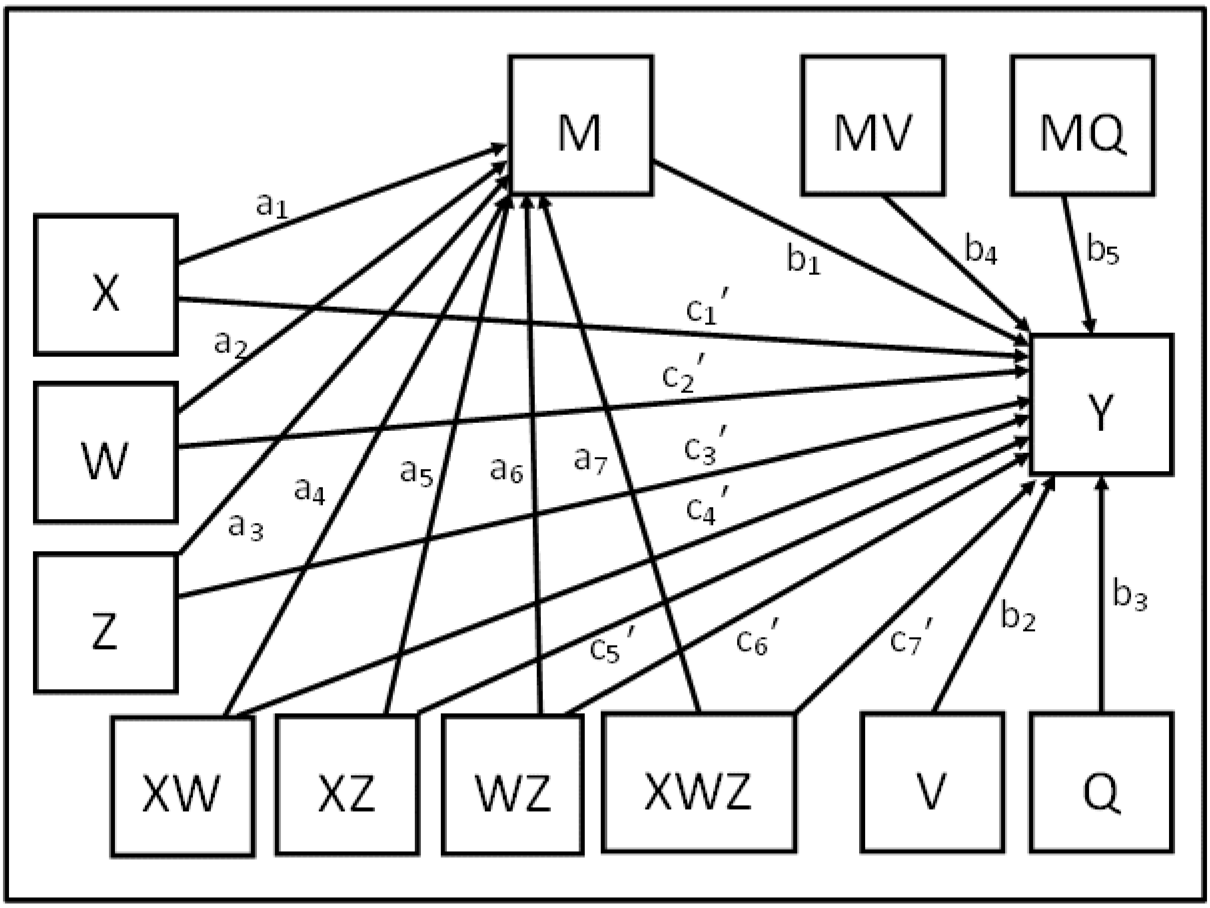
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2V + b3Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + b5(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)Q + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4ZV + a4b4XWV + a5b4XZV + a6b4WZV + a7b4XWZV + a0b5Q + a1b5XQ + a2b5WQ + a3b5ZQ + a4b5XWQ + a5b5XZQ + a6b5WZQ + a7b5XWZQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + b2V + b3Q + a0b4V + a2b4WV + a3b4ZV + a6b4WZV + a0b5Q + a2b5WQ + a3b5ZQ + a6b5WZQ + c2'W + c3'Z + c6'WZ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4V + a4b4WV + a5b4ZV + a7b4WZV + a1b5Q + a4b5WQ + a5b5ZQ + a7b5WZQ + c1' + c4'W + c5'Z + c7'WZ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4V + a4b4WV + a5b4ZV + a7b4WZV + a1b5Q + a4b5WQ + a5b5ZQ + a7b5WZQ = (a1 + a4W + a5Z + a7WZ) (b1 + b4V + b5Q)

One direct effect of X on Y, conditional on W, Z:

c1' + c4'W + c5'Z + c7'WZ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ WZ MV MQ XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;
```

```
ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON Z (cdash3);
   Y ON XW (cdash4);
   Y ON XZ (cdash5);
   Y ON WZ (cdash6);
   Y ON XWZ (cdash7);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
```

```
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values
```

```
    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q;

    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
```

```
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q;

    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
```

```
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q;
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q;

    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
```

```
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*MED_Q +
```

```
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q;
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q;

    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q;

    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
```

```
      a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
      a7*b4*MED_W*LOW_Z*MED_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
      a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q;
     IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
      a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
      a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
      a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q;

     ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
      a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
      a7*b4*LOW_W*MED_Z*MED_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
      a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q;
     IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
      a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
      a7*b4*MED_W*MED_Z*MED_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
      a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q;
     IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
      a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
      a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
      a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q;

     ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
      a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
      a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
      a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q;
     IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
      a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
      a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
      a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q;
     IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
      a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
      a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
      a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q;
```

```
    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
```

```
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q;

    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
```

```
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q;
    IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q;

    ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q;
    IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q;
    IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q;

    ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q;
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
```

```
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
```

```
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q;

    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values
```

```
    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + DLOW_LOZ;
    TMLLL = IMLLL + DMEW_LOZ;
    THLLL = IHLLL + DHIW_LOZ;

    TLMLL = ILMLL + DLOW_MEZ;
    TMMLL = IMMLL + DMEW_MEZ;
    THMLL = IHMLL + DHIW_MEZ;

    TLHLL = ILHLL + DLOW_HIZ;
    TMHLL = IMHLL + DMEW_HIZ;
    THHLL = IHHLL + DHIW_HIZ;

    TLLML = ILLML + DLOW_LOZ;
    TMLML = IMLML + DMEW_LOZ;
    THLML = IHLML + DHIW_LOZ;

    TLMML = ILMML + DLOW_MEZ;
    TMMML = IMMML + DMEW_MEZ;
    THMML = IHMML + DHIW_MEZ;

    TLHML = ILHML + DLOW_HIZ;
    TMHML = IMHML + DMEW_HIZ;
    THHML = IHHML + DHIW_HIZ;

    TLLHL = ILLHL + DLOW_LOZ;
    TMLHL = IMLHL + DMEW_LOZ;
    THLHL = IHLHL + DHIW_LOZ;
```

```
TLMHL = ILMHL + DLOW_MEZ;
TMMHL = IMMHL + DMEW_MEZ;
THMHL = IHMHL + DHIW_MEZ;

TLHHL = ILHHL + DLOW_HIZ;
TMHHL = IMHHL + DMEW_HIZ;
THHHL = IHHHL + DHIW_HIZ;

TLLLM = ILLLM + DLOW_LOZ;
TMLLM = IMLLM + DMEW_LOZ;
THLLM = IHLLM + DHIW_LOZ;

TLMLM = ILMLM + DLOW_MEZ;
TMMLM = IMMLM + DMEW_MEZ;
THMLM = IHMLM + DHIW_MEZ;

TLHLM = ILHLM + DLOW_HIZ;
TMHLM = IMHLM + DMEW_HIZ;
THHLM = IHHLM + DHIW_HIZ;

TLLMM = ILLMM + DLOW_LOZ;
TMLMM = IMLMM + DMEW_LOZ;
THLMM = IHLMM + DHIW_LOZ;

TLMMM = ILMMM + DLOW_MEZ;
TMMMM = IMMMM + DMEW_MEZ;
THMMM = IHMMM + DHIW_MEZ;

TLHMM = ILHMM + DLOW_HIZ;
TMHMM = IMHMM + DMEW_HIZ;
THHMM = IHHMM + DHIW_HIZ;

TLLHM = ILLHM + DLOW_LOZ;
TMLHM = IMLHM + DMEW_LOZ;
THLHM = IHLHM + DHIW_LOZ;

TLMHM = ILMHM + DLOW_MEZ;
TMMHM = IMMHM + DMEW_MEZ;
THMHM = IHMHM + DHIW_MEZ;

TLHHM = ILHHM + DLOW_HIZ;
TMHHM = IMHHM + DMEW_HIZ;
THHHM = IHHHM + DHIW_HIZ;

TLLLH = ILLLH + DLOW_LOZ;
TMLLH = IMLLH + DMEW_LOZ;
THLLH = IHLLH + DHIW_LOZ;

TLMLH = ILMLH + DLOW_MEZ;
TMMLH = IMMLH + DMEW_MEZ;
THMLH = IHMLH + DHIW_MEZ;
```

```
    TLHLH = ILHLH + DLOW_HIZ;
    TMHLH = IMHLH + DMEW_HIZ;
    THHLH = IHHLH + DHIW_HIZ;

    TLLMH = ILLMH + DLOW_LOZ;
    TMLMH = IMLMH + DMEW_LOZ;
    THLMH = IHLMH + DHIW_LOZ;

    TLMMH = ILMMH + DLOW_MEZ;
    TMMMH = IMMMH + DMEW_MEZ;
    THMMH = IHMMH + DHIW_MEZ;

    TLHMH = ILHMH + DLOW_HIZ;
    TMHMH = IMHMH + DMEW_HIZ;
    THHMH = IHHMH + DHIW_HIZ;

    TLLHH = ILLHH + DLOW_LOZ;
    TMLHH = IMLHH + DMEW_LOZ;
    THLHH = IHLHH + DHIW_LOZ;

    TLMHH = ILMHH + DLOW_MEZ;
    TMMHH = IMMHH + DMEW_MEZ;
    THMHH = IHMHH + DHIW_MEZ;

    TLHHH = ILHHH + DLOW_HIZ;
    TMHHH = IMHHH + DMEW_HIZ;
    THHHH = IHHHH + DHIW_HIZ;
! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;
```

```
PLMLL = ILMLL*XVAL;
PMMLL = IMMLL*XVAL;
PHMLL = IHMLL*XVAL;

PLHLL = ILHLL*XVAL;
PMHLL = IMHLL*XVAL;
PHHLL = IHHLL*XVAL;

PLLML = ILLML*XVAL;
PMLML = IMLML*XVAL;
PHLML = IHLML*XVAL;

PLMML = ILMML*XVAL;
PMMML = IMMML*XVAL;
PHMML = IHMML*XVAL;

PLHML = ILHML*XVAL;
PMHML = IMHML*XVAL;
PHHML = IHHML*XVAL;

PLLHL = ILLHL*XVAL;
PMLHL = IMLHL*XVAL;
PHLHL = IHLHL*XVAL;

PLMHL = ILMHL*XVAL;
PMMHL = IMMHL*XVAL;
PHMHL = IHMHL*XVAL;

PLHHL = ILHHL*XVAL;
PMHHL = IMHHL*XVAL;
PHHHL = IHHHL*XVAL;

PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;
```

```
PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;

PLHLH = ILHLH*XVAL;
PMHLH = IMHLH*XVAL;
PHHLH = IHHLH*XVAL;

PLLMH = ILLMH*XVAL;
PMLMH = IMLMH*XVAL;
PHLMH = IHLMH*XVAL;

PLMMH = ILMMH*XVAL;
PMMMH = IMMMH*XVAL;
PHMMH = IHMMH*XVAL;

PLHMH = ILHMH*XVAL;
PMHMH = IMHMH*XVAL;
PHHMH = IHHMH*XVAL;

PLLHH = ILLHH*XVAL;
PMLHH = IMLHH*XVAL;
PHLHH = IHLHH*XVAL;

PLMHH = ILMHH*XVAL;
PMMHH = IMMHH*XVAL;
PHMHH = IHMHH*XVAL;

PLHHH = ILHHH*XVAL;
PMHHH = IMHHH*XVAL;
PHHHH = IHHHH*XVAL;
```

```
PLOT:
   TYPE = plot2;
OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 52: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path with all 2-way and 3-way interactions, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path

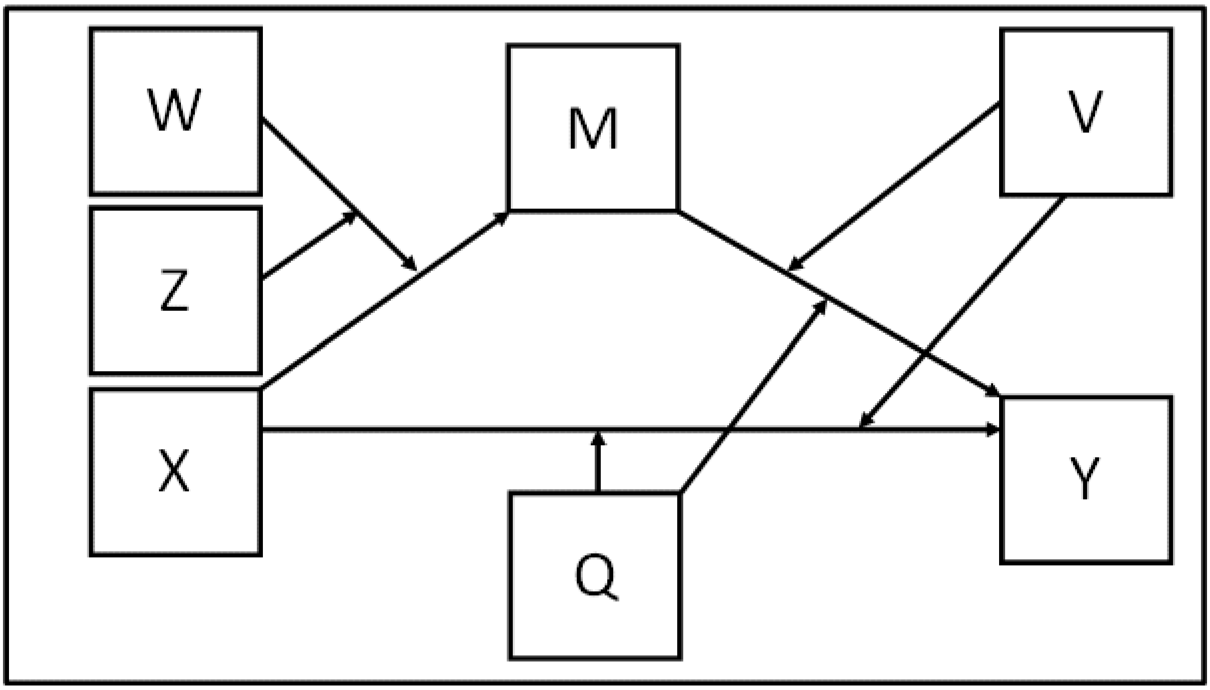
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

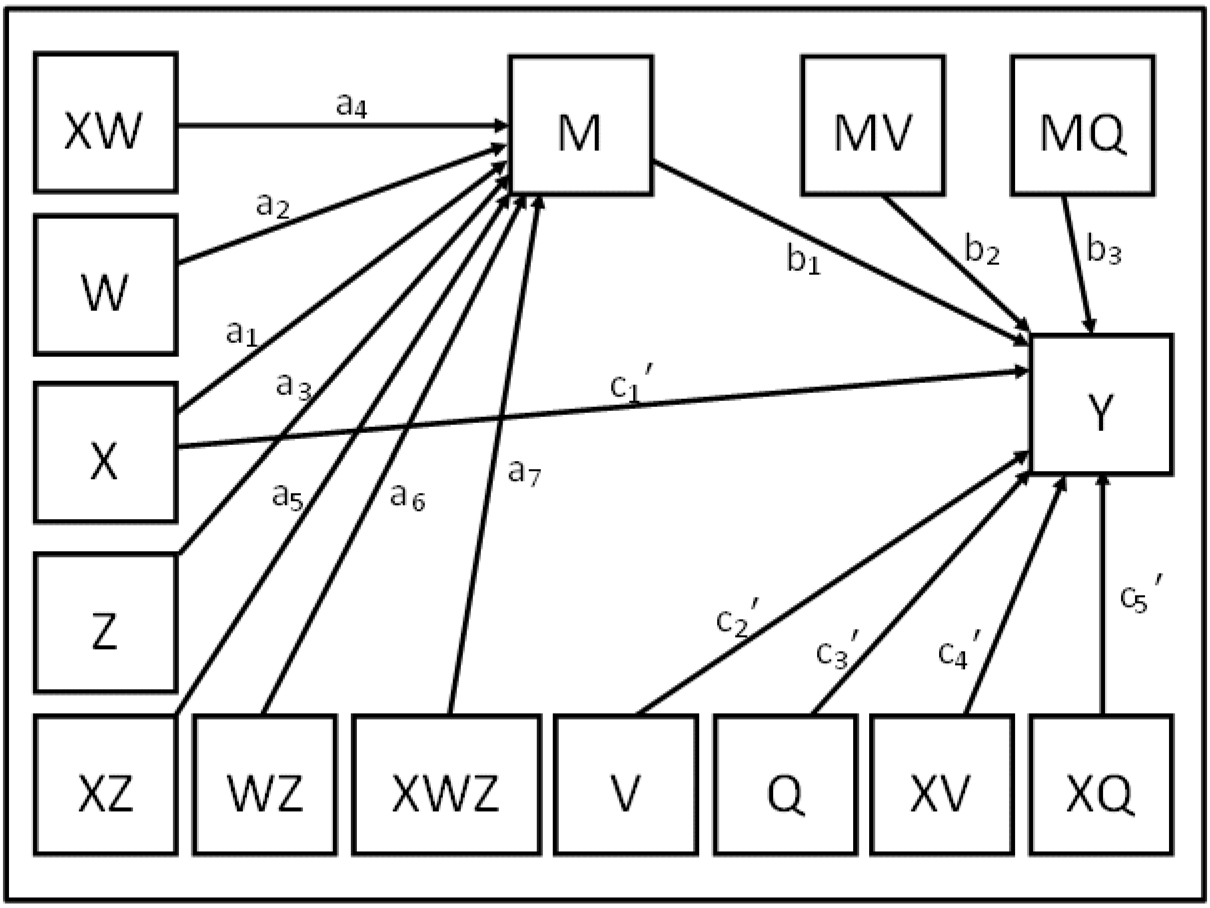
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)Q + c1'X + c2'V + c3'Q + c4'XV + c5'XQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + a0b2V + a1b2XV + a2b2WV + a3b2ZV + a4b2XWV + a5b2XZV + a6b2WZV + a7b2XWZV + a0b3Q + a1b3XQ + a2b3WQ + a3b3ZQ + a4b3XWQ + a5b3XZQ + a6b3WZQ + a7b3XWZQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + a0b2V + a2b2WV + a3b2ZV + a6b2WZV + a0b3Q + a2b3WQ + a3b3ZQ + a6b3WZQ + c2'V + c3'Q) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV + a1b3Q + a4b3WQ + a5b3ZQ + a7b3WZQ + c1' + c4'V + c5'Q)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV + a1b3Q + a4b3WQ + a5b3ZQ + a7b3WZQ = (a1 + a4W + a5Z + a7WZ)(b1 + b2V + b3Q)

One direct effect of X on Y, conditional on V, Q:

c1' + c4'V + c5'Q

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ WZ XV XQ MV MQ XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   XQ = X*Q;
   XV = X*V;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
```

```
    ESTIMATOR = ML;
    BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON MV (b2);
    Y ON MQ (b3);

    Y ON X (cdash1);
    Y ON V (cdash2);
    Y ON Q (cdash3);
    Y ON XV (cdash4);
    Y ON XQ (cdash5);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
    M ON WZ (a6);
    M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
     NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
     ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
     ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
     ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
     ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
     ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
     ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
     ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
     ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
```

```
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    DLOV_LOQ DMEV_LOQ DHIV_LOQ DLOV_MEQ DMEV_MEQ DHIV_MEQ
    DLOV_HIQ DMEV_HIQ DHIV_HIQ
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*LOW_W*LOW_Z*LOW_Q;
```

```
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*MED_W*LOW_Z*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*HIGH_W*LOW_Z*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*LOW_W*MED_Z*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*MED_W*MED_Z*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*HIGH_W*MED_Z*LOW_Q;

    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*LOW_W*HIGH_Z*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*MED_W*HIGH_Z*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V + a1*b3*LOW_Q +
```

```
a4*b3*HIGH_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*HIGH_W*HIGH_Z*LOW_Q;

    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*LOW_W*LOW_Z*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*MED_W*LOW_Z*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*HIGH_W*LOW_Z*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*LOW_W*MED_Z*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*MED_W*MED_Z*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*HIGH_W*MED_Z*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*LOW_W*HIGH_Z*LOW_Q;
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
```

```
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*MED_W*HIGH_Z*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*HIGH_W*HIGH_Z*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*LOW_W*LOW_Z*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*MED_W*LOW_Z*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*HIGH_W*LOW_Z*LOW_Q;

    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*LOW_W*MED_Z*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*MED_W*MED_Z*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V + a1*b3*LOW_Q +
```

```
a4*b3*HIGH_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*HIGH_W*MED_Z*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*LOW_W*HIGH_Z*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*MED_W*HIGH_Z*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*HIGH_W*HIGH_Z*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*LOW_W*LOW_Z*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*MED_W*LOW_Z*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*HIGH_W*LOW_Z*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
```

```
     a5*b3*MED_Z*MED_Q + a7*b3*LOW_W*MED_Z*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*MED_W*MED_Z*MED_Q;
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*HIGH_W*MED_Z*MED_Q;

    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*LOW_W*HIGH_Z*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*MED_W*HIGH_Z*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*HIGH_W*HIGH_Z*MED_Q;

    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*LOW_W*LOW_Z*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*MED_W*LOW_Z*MED_Q;
    IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
```

```
     a7*b2*HIGH_W*LOW_Z*MED_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*HIGH_W*LOW_Z*MED_Q;

    ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*LOW_W*MED_Z*MED_Q;
    IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*MED_W*MED_Z*MED_Q;
    IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*HIGH_W*MED_Z*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*LOW_W*HIGH_Z*MED_Q;
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*MED_W*HIGH_Z*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*HIGH_W*HIGH_Z*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*LOW_W*LOW_Z*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
```

```
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*MED_W*LOW_Z*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*HIGH_W*LOW_Z*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*LOW_W*MED_Z*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*MED_W*MED_Z*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*HIGH_W*MED_Z*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*LOW_W*HIGH_Z*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*MED_W*HIGH_Z*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
```

```
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*HIGH_W*HIGH_Z*MED_Q;

    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*LOW_W*LOW_Z*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*MED_W*LOW_Z*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*HIGH_W*LOW_Z*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*LOW_W*MED_Z*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*MED_W*MED_Z*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*HIGH_W*MED_Z*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
```

```
a4*b3*LOW_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*LOW_W*HIGH_Z*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*MED_W*HIGH_Z*HIGH_Q;
    IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*HIGH_W*HIGH_Z*HIGH_Q;

    ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*LOW_W*LOW_Z*HIGH_Q;
    IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*MED_W*LOW_Z*HIGH_Q;
    IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*HIGH_W*LOW_Z*HIGH_Q;

    ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*LOW_W*MED_Z*HIGH_Q;
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*MED_W*MED_Z*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
```

```
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*HIGH_W*MED_Z*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*LOW_W*HIGH_Z*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*MED_W*HIGH_Z*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*HIGH_W*HIGH_Z*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*LOW_W*LOW_Z*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*MED_W*LOW_Z*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*HIGH_W*LOW_Z*HIGH_Q;

    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
```

```
     a5*b3*MED_Z*HIGH_Q + a7*b3*LOW_W*MED_Z*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*MED_W*MED_Z*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*HIGH_W*MED_Z*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*LOW_W*HIGH_Z*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*MED_W*HIGH_Z*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*HIGH_W*HIGH_Z*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOV_LOQ = cdash1 + cdash4*LOW_V + cdash5*LOW_Q;
    DMEV_LOQ = cdash1 + cdash4*MED_W + cdash5*LOW_Q;
    DHIV_LOQ = cdash1 + cdash4*HIGH_V + cdash5*LOW_Q;

    DLOV_MEQ = cdash1 + cdash4*LOW_V + cdash5*MED_Q;
    DMEV_MEQ = cdash1 + cdash4*MED_W + cdash5*MED_Q;
    DHIV_MEQ = cdash1 + cdash4*HIGH_V + cdash5*MED_Q;
```

```
    DLOV_HIQ = cdash1 + cdash4*LOW_V + cdash5*HIGH_Q;
    DMEV_HIQ = cdash1 + cdash4*MED_W + cdash5*HIGH_Q;
    DHIV_HIQ = cdash1 + cdash4*HIGH_V + cdash5*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + DLOV_LOQ;
    TMLLL = IMLLL + DLOV_LOQ;
    THLLL = IHLLL + DLOV_LOQ;

    TLMLL = ILMLL + DLOV_LOQ;
    TMMLL = IMMLL + DLOV_LOQ;
    THMLL = IHMLL + DLOV_LOQ;

    TLHLL = ILHLL + DLOV_LOQ;
    TMHLL = IMHLL + DLOV_LOQ;
    THHLL = IHHLL + DLOV_LOQ;

    TLLML = ILLML + DMEV_LOQ;
    TMLML = IMLML + DMEV_LOQ;
    THLML = IHLML + DMEV_LOQ;

    TLMML = ILMML + DMEV_LOQ;
    TMMML = IMMML + DMEV_LOQ;
    THMML = IHMML + DMEV_LOQ;

    TLHML = ILHML + DMEV_LOQ;
    TMHML = IMHML + DMEV_LOQ;
    THHML = IHHML + DMEV_LOQ;

    TLLHL = ILLHL + DHIV_LOQ;
    TMLHL = IMLHL + DHIV_LOQ;
    THLHL = IHLHL + DHIV_LOQ;

    TLMHL = ILMHL + DHIV_LOQ;
    TMMHL = IMMHL + DHIV_LOQ;
    THMHL = IHMHL + DHIV_LOQ;

    TLHHL = ILHHL + DHIV_LOQ;
    TMHHL = IMHHL + DHIV_LOQ;
    THHHL = IHHHL + DHIV_LOQ;

    TLLLM = ILLLM + DLOV_MEQ;
    TMLLM = IMLLM + DLOV_MEQ;
    THLLM = IHLLM + DLOV_MEQ;

    TLMLM = ILMLM + DLOV_MEQ;
    TMMLM = IMMLM + DLOV_MEQ;
    THMLM = IHMLM + DLOV_MEQ;
```

```
TLHLM = ILHLM + DLOV_MEQ;
TMHLM = IMHLM + DLOV_MEQ;
THHLM = IHHLM + DLOV_MEQ;

TLLMM = ILLMM + DMEV_MEQ;
TMLMM = IMLMM + DMEV_MEQ;
THLMM = IHLMM + DMEV_MEQ;

TLMMM = ILMMM + DMEV_MEQ;
TMMMM = IMMMM + DMEV_MEQ;
THMMM = IHMMM + DMEV_MEQ;

TLHMM = ILHMM + DMEV_MEQ;
TMHMM = IMHMM + DMEV_MEQ;
THHMM = IHHMM + DMEV_MEQ;

TLLHM = ILLHM + DHIV_MEQ;
TMLHM = IMLHM + DHIV_MEQ;
THLHM = IHLHM + DHIV_MEQ;

TLMHM = ILMHM + DHIV_MEQ;
TMMHM = IMMHM + DHIV_MEQ;
THMHM = IHMHM + DHIV_MEQ;

TLHHM = ILHHM + DHIV_MEQ;
TMHHM = IMHHM + DHIV_MEQ;
THHHM = IHHHM + DHIV_MEQ;

TLLLH = ILLLH + DLOV_HIQ;
TMLLH = IMLLH + DLOV_HIQ;
THLLH = IHLLH + DLOV_HIQ;

TLMLH = ILMLH + DLOV_HIQ;
TMMLH = IMMLH + DLOV_HIQ;
THMLH = IHMLH + DLOV_HIQ;

TLHLH = ILHLH + DLOV_HIQ;
TMHLH = IMHLH + DLOV_HIQ;
THHLH = IHHLH + DLOV_HIQ;

TLLMH = ILLMH + DMEV_HIQ;
TMLMH = IMLMH + DMEV_HIQ;
THLMH = IHLMH + DMEV_HIQ;

TLMMH = ILMMH + DMEV_HIQ;
TMMMH = IMMMH + DMEV_HIQ;
THMMH = IHMMH + DMEV_HIQ;

TLHMH = ILHMH + DMEV_HIQ;
TMHMH = IMHMH + DMEV_HIQ;
THHMH = IHHMH + DMEV_HIQ;
```

```
    TLLHH = ILLHH + DHIV_HIQ;
    TMLHH = IMLHH + DHIV_HIQ;
    THLHH = IHLHH + DHIV_HIQ;

    TLMHH = ILMHH + DHIV_HIQ;
    TMMHH = IMMHH + DHIV_HIQ;
    THMHH = IHMHH + DHIV_HIQ;

    TLHHH = ILHHH + DHIV_HIQ;
    TMHHH = IMHHH + DHIV_HIQ;
    THHHH = IHHHH + DHIV_HIQ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;

    PLMLL = ILMLL*XVAL;
    PMMLL = IMMLL*XVAL;
    PHMLL = IHMLL*XVAL;

    PLHLL = ILHLL*XVAL;
    PMHLL = IMHLL*XVAL;
    PHHLL = IHHLL*XVAL;

    PLLML = ILLML*XVAL;
    PMLML = IMLML*XVAL;
    PHLML = IHLML*XVAL;

    PLMML = ILMML*XVAL;
    PMMML = IMMML*XVAL;
    PHMML = IHMML*XVAL;
```

```
PLHML = ILHML*XVAL;
PMHML = IMHML*XVAL;
PHHML = IHHML*XVAL;

PLLHL = ILLHL*XVAL;
PMLHL = IMLHL*XVAL;
PHLHL = IHLHL*XVAL;

PLMHL = ILMHL*XVAL;
PMMHL = IMMHL*XVAL;
PHMHL = IHMHL*XVAL;

PLHHL = ILHHL*XVAL;
PMHHL = IMHHL*XVAL;
PHHHL = IHHHL*XVAL;

PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;
```

```
    PLLLH = ILLLH*XVAL;
    PMLLH = IMLLH*XVAL;
    PHLLH = IHLLH*XVAL;

    PLMLH = ILMLH*XVAL;
    PMMLH = IMMLH*XVAL;
    PHMLH = IHMLH*XVAL;

    PLHLH = ILHLH*XVAL;
    PMHLH = IMHLH*XVAL;
    PHHLH = IHHLH*XVAL;

    PLLMH = ILLMH*XVAL;
    PMLMH = IMLMH*XVAL;
    PHLMH = IHLMH*XVAL;

    PLMMH = ILMMH*XVAL;
    PMMMH = IMMMH*XVAL;
    PHMMH = IHMMH*XVAL;

    PLHMH = ILHMH*XVAL;
    PMHMH = IMHMH*XVAL;
    PHHMH = IHHMH*XVAL;

    PLLHH = ILLHH*XVAL;
    PMLHH = IMLHH*XVAL;
    PHLHH = IHLHH*XVAL;

    PLMHH = ILMHH*XVAL;
    PMMHH = IMMHH*XVAL;
    PHMHH = IHMHH*XVAL;

    PLHHH = ILHHH*XVAL;
    PMHHH = IMHHH*XVAL;
    PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 53: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path, with the other 2 moderating the Mediator-DV path with all 2-way and 3-way interactions

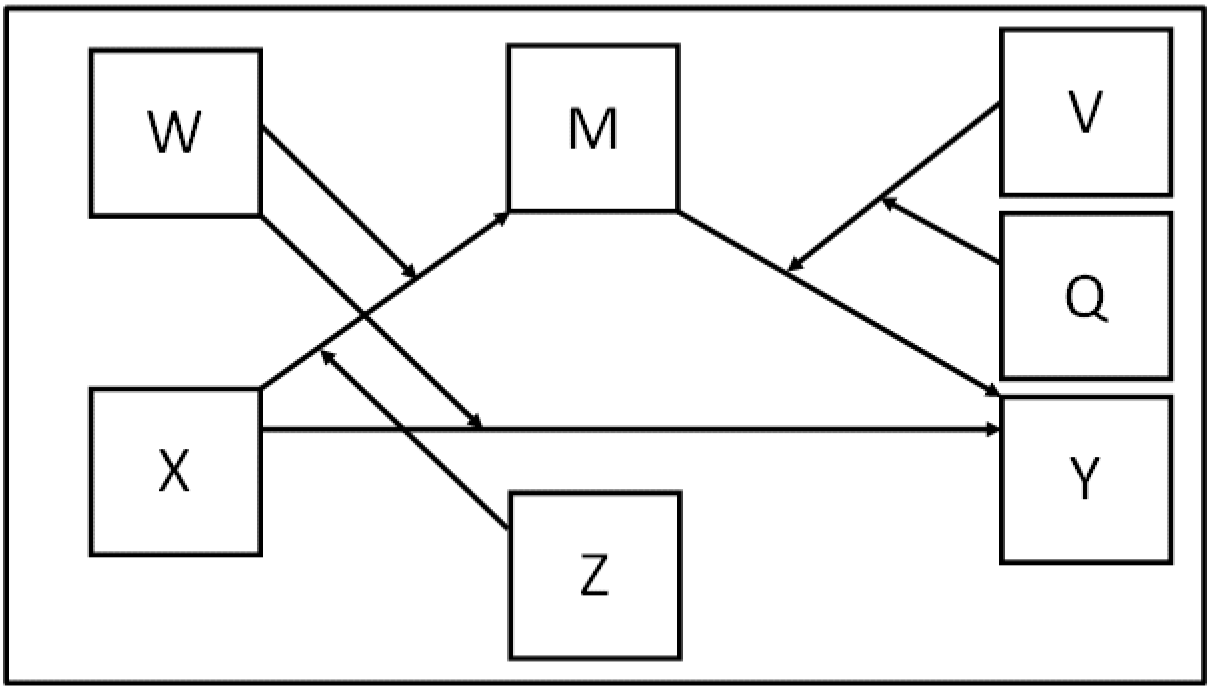
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

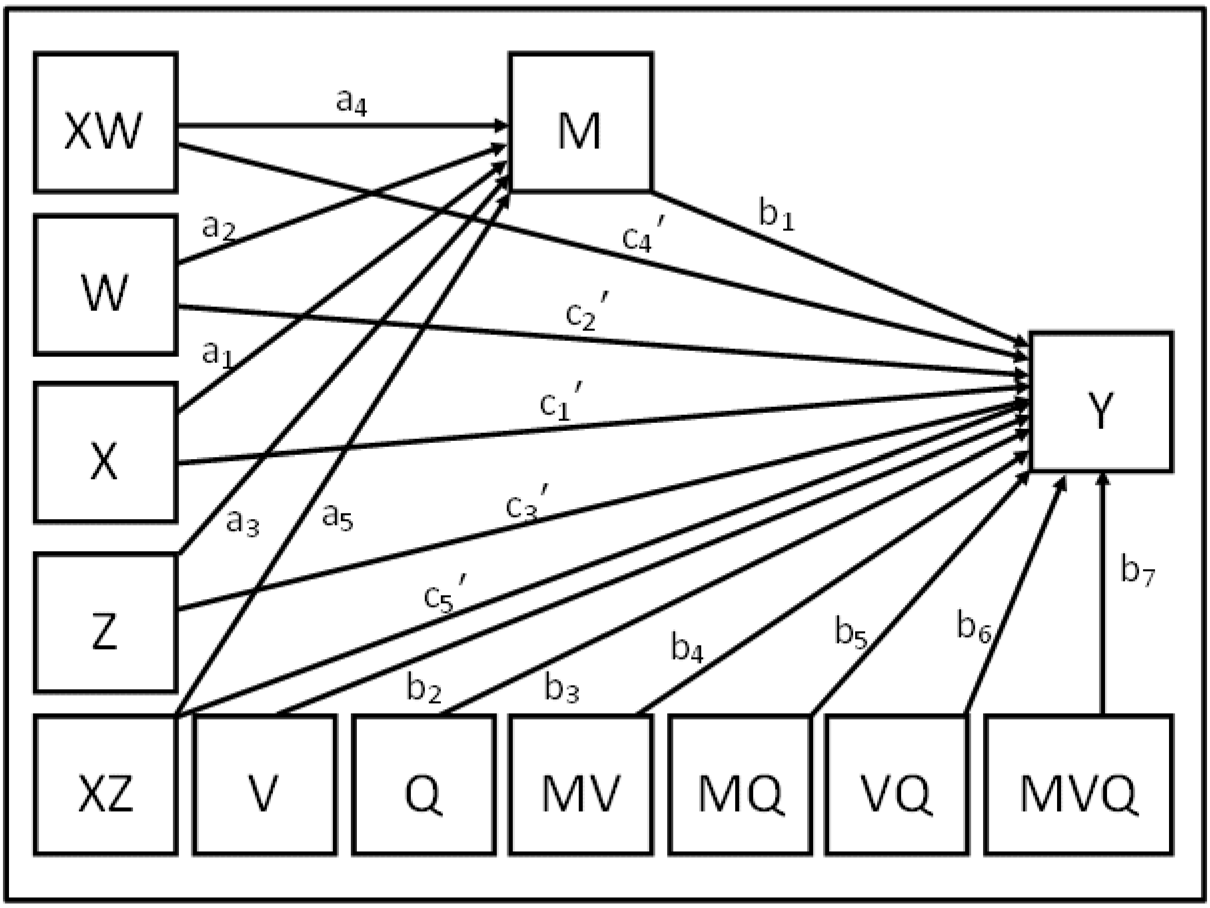
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2V + b3Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ)V + b5(a0 + a1X + a2W + a3Z + a4XW + a5XZ)Q + b6VQ + b7(a0 + a1X + a2W + a3Z + a4XW + a5XZ)VQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4ZV + a4b4XWV + a5b4XZV + a0b5Q + a1b5XQ + a2b5WQ + a3b5ZQ + a4b5XWQ + a5b5XZQ + b6VQ + a0b7VQ + a1b7XVQ + a2b7WVQ + a3b7ZVQ + a4b7XWVQ + a5b7XZVQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + b2V + b3Q + a0b4V + a2b4WV + a3b4ZV + a0b5Q + a2b5WQ + a3b5ZQ + b6VQ + a0b7VQ + a2b7WVQ + a3b7ZVQ + c2'W + c3'Z) + (a1b1 + a4b1W + a5b1Z + a1b4V + a4b4WV + a5b4ZV + a1b5Q + a4b5WQ + a5b5ZQ + a1b7VQ + a4b7WVQ + a5b7ZVQ + c1' + c4'W + c5'Z)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a1b4V + a4b4WV + a5b4ZV + a1b5Q + a4b5WQ + a5b5ZQ + a1b7VQ + a4b7WVQ + a5b7ZVQ = (a1 + a4W + a5Z)(b1 + b4V + b5Q + b7VQ)

One direct effect of X on Y, conditional on W, Z:

c1' + c4'W + c5'Z

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ VQ MV MQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   VQ = V*Q;
   MVQ = M*V*Q;
```

```
ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);
   Y ON VQ (b6);
   Y ON MVQ (b7);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON Z (cdash3);
   Y ON XW (cdash4);
   Y ON XZ (cdash5);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
```

```
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*LOW_Q +
```

```
     a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q + a1*b7*LOW_V*LOW_Q
+
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q + a1*b7*LOW_V*LOW_Q
+
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q + a1*b7*LOW_V*LOW_Q
+
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q + a1*b7*LOW_V*LOW_Q
+
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q;

    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
```

```
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q;

    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q + a1*b7*MED_V*LOW_Q
+
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q + a1*b7*MED_V*LOW_Q
+
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q + a1*b7*MED_V*LOW_Q
+
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q + a1*b7*MED_V*LOW_Q
+
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q;
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q;
```

```
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q;

    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*LOW_Q +
     a4*b5*HIGH_W*LOW_Q + a5*b5*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
     a4*b5*LOW_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
```

```
a1*b7*HIGH_V*LOW_Q +
     a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
     a4*b5*MED_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*LOW_Q
+
     a4*b5*HIGH_W*LOW_Q + a5*b5*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q + a1*b7*LOW_V*MED_Q
+
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q + a1*b7*LOW_V*MED_Q
+
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q + a1*b7*LOW_V*MED_Q
+
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q + a1*b7*LOW_V*MED_Q
+
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q;
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q +
```

```
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q;

    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q;

    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q + a1*b7*MED_V*MED_Q
+
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q + a1*b7*MED_V*MED_Q
+
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q;
    IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q;

    ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q + a1*b7*MED_V*MED_Q
+
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q;
    IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q + a1*b7*MED_V*MED_Q
+
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q;
```

```
    IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q;
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*MED_Z*MED_Q +
```

```
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*HIGH_W*MED_Q + a5*b5*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*LOW_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*MED_Q +
     a4*b5*MED_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*MED_Q
+
     a4*b5*HIGH_W*MED_Q + a5*b5*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q;

    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V + a1*b5*HIGH_Q +
```

```
     a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*LOW_V +
     a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q;
    IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*LOW_V
+
     a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q;

    ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q;
    IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
```

```
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q;
    IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q;

    ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q;
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_V +
     a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*MED_V
+
     a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V + a1*b5*HIGH_Q +
     a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
```

```
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q;

    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*HIGH_Q +
     a4*b5*LOW_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*HIGH_Q +
     a4*b5*MED_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*LOW_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*MED_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
```

```
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b4*HIGH_V
+
     a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V + a1*b5*HIGH_Q
+
     a4*b5*HIGH_W*HIGH_Q + a5*b5*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + DLOW_LOZ;
    TMLLL = IMLLL + DMEW_LOZ;
    THLLL = IHLLL + DHIW_LOZ;

    TLMLL = ILMLL + DLOW_MEZ;
    TMMLL = IMMLL + DMEW_MEZ;
    THMLL = IHMLL + DHIW_MEZ;

    TLHLL = ILHLL + DLOW_HIZ;
    TMHLL = IMHLL + DMEW_HIZ;
    THHLL = IHHLL + DHIW_HIZ;

    TLLML = ILLML + DLOW_LOZ;
    TMLML = IMLML + DMEW_LOZ;
    THLML = IHLML + DHIW_LOZ;

    TLMML = ILMML + DLOW_MEZ;
    TMMML = IMMML + DMEW_MEZ;
    THMML = IHMML + DHIW_MEZ;

    TLHML = ILHML + DLOW_HIZ;
    TMHML = IMHML + DMEW_HIZ;
    THHML = IHHML + DHIW_HIZ;
```

```
TLLHL = ILLHL + DLOW_LOZ;
TMLHL = IMLHL + DMEW_LOZ;
THLHL = IHLHL + DHIW_LOZ;

TLMHL = ILMHL + DLOW_MEZ;
TMMHL = IMMHL + DMEW_MEZ;
THMHL = IHMHL + DHIW_MEZ;

TLHHL = ILHHL + DLOW_HIZ;
TMHHL = IMHHL + DMEW_HIZ;
THHHL = IHHHL + DHIW_HIZ;

TLLLM = ILLLM + DLOW_LOZ;
TMLLM = IMLLM + DMEW_LOZ;
THLLM = IHLLM + DHIW_LOZ;

TLMLM = ILMLM + DLOW_MEZ;
TMMLM = IMMLM + DMEW_MEZ;
THMLM = IHMLM + DHIW_MEZ;

TLHLM = ILHLM + DLOW_HIZ;
TMHLM = IMHLM + DMEW_HIZ;
THHLM = IHHLM + DHIW_HIZ;

TLLMM = ILLMM + DLOW_LOZ;
TMLMM = IMLMM + DMEW_LOZ;
THLMM = IHLMM + DHIW_LOZ;

TLMMM = ILMMM + DLOW_MEZ;
TMMMM = IMMMM + DMEW_MEZ;
THMMM = IHMMM + DHIW_MEZ;

TLHMM = ILHMM + DLOW_HIZ;
TMHMM = IMHMM + DMEW_HIZ;
THHMM = IHHMM + DHIW_HIZ;

TLLHM = ILLHM + DLOW_LOZ;
TMLHM = IMLHM + DMEW_LOZ;
THLHM = IHLHM + DHIW_LOZ;

TLMHM = ILMHM + DLOW_MEZ;
TMMHM = IMMHM + DMEW_MEZ;
THMHM = IHMHM + DHIW_MEZ;

TLHHM = ILHHM + DLOW_HIZ;
TMHHM = IMHHM + DMEW_HIZ;
THHHM = IHHHM + DHIW_HIZ;

TLLLH = ILLLH + DLOW_LOZ;
TMLLH = IMLLH + DMEW_LOZ;
THLLH = IHLLH + DHIW_LOZ;
```

```
    TLMLH = ILMLH + DLOW_MEZ;
    TMMLH = IMMLH + DMEW_MEZ;
    THMLH = IHMLH + DHIW_MEZ;

    TLHLH = ILHLH + DLOW_HIZ;
    TMHLH = IMHLH + DMEW_HIZ;
    THHLH = IHHLH + DHIW_HIZ;

    TLLMH = ILLMH + DLOW_LOZ;
    TMLMH = IMLMH + DMEW_LOZ;
    THLMH = IHLMH + DHIW_LOZ;

    TLMMH = ILMMH + DLOW_MEZ;
    TMMMH = IMMMH + DMEW_MEZ;
    THMMH = IHMMH + DHIW_MEZ;

    TLHMH = ILHMH + DLOW_HIZ;
    TMHMH = IMHMH + DMEW_HIZ;
    THHMH = IHHMH + DHIW_HIZ;

    TLLHH = ILLHH + DLOW_LOZ;
    TMLHH = IMLHH + DMEW_LOZ;
    THLHH = IHLHH + DHIW_LOZ;

    TLMHH = ILMHH + DLOW_MEZ;
    TMMHH = IMMHH + DMEW_MEZ;
    THMHH = IHMHH + DHIW_MEZ;

    TLHHH = ILHHH + DLOW_HIZ;
    TMHHH = IMHHH + DMEW_HIZ;
    THHHH = IHHHH + DHIW_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);
```

```
PLLLL = ILLLL*XVAL;
PMLLL = IMLLL*XVAL;
PHLLL = IHLLL*XVAL;

PLMLL = ILMLL*XVAL;
PMMLL = IMMLL*XVAL;
PHMLL = IHMLL*XVAL;

PLHLL = ILHLL*XVAL;
PMHLL = IMHLL*XVAL;
PHHLL = IHHLL*XVAL;

PLLML = ILLML*XVAL;
PMLML = IMLML*XVAL;
PHLML = IHLML*XVAL;

PLMML = ILMML*XVAL;
PMMML = IMMML*XVAL;
PHMML = IHMML*XVAL;

PLHML = ILHML*XVAL;
PMHML = IMHML*XVAL;
PHHML = IHHML*XVAL;

PLLHL = ILLHL*XVAL;
PMLHL = IMLHL*XVAL;
PHLHL = IHLHL*XVAL;

PLMHL = ILMHL*XVAL;
PMMHL = IMMHL*XVAL;
PHMHL = IHMHL*XVAL;

PLHHL = ILHHL*XVAL;
PMHHL = IMHHL*XVAL;
PHHHL = IHHHL*XVAL;

PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;
```

```
PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;

PLHLH = ILHLH*XVAL;
PMHLH = IMHLH*XVAL;
PHHLH = IHHLH*XVAL;

PLLMH = ILLMH*XVAL;
PMLMH = IMLMH*XVAL;
PHLMH = IHLMH*XVAL;

PLMMH = ILMMH*XVAL;
PMMMH = IMMMH*XVAL;
PHMMH = IHMMH*XVAL;

PLHMH = ILHMH*XVAL;
PMHMH = IMHMH*XVAL;
PHHMH = IHHMH*XVAL;

PLLHH = ILLHH*XVAL;
PMLHH = IMLHH*XVAL;
PHLHH = IHLHH*XVAL;

PLMHH = ILMHH*XVAL;
PMMHH = IMMHH*XVAL;
PHMHH = IHMHH*XVAL;
```

```
     PLHHH = ILHHH*XVAL;
     PMHHH = IMHHH*XVAL;
     PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 54: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path with all 2-way and 3-way interactions

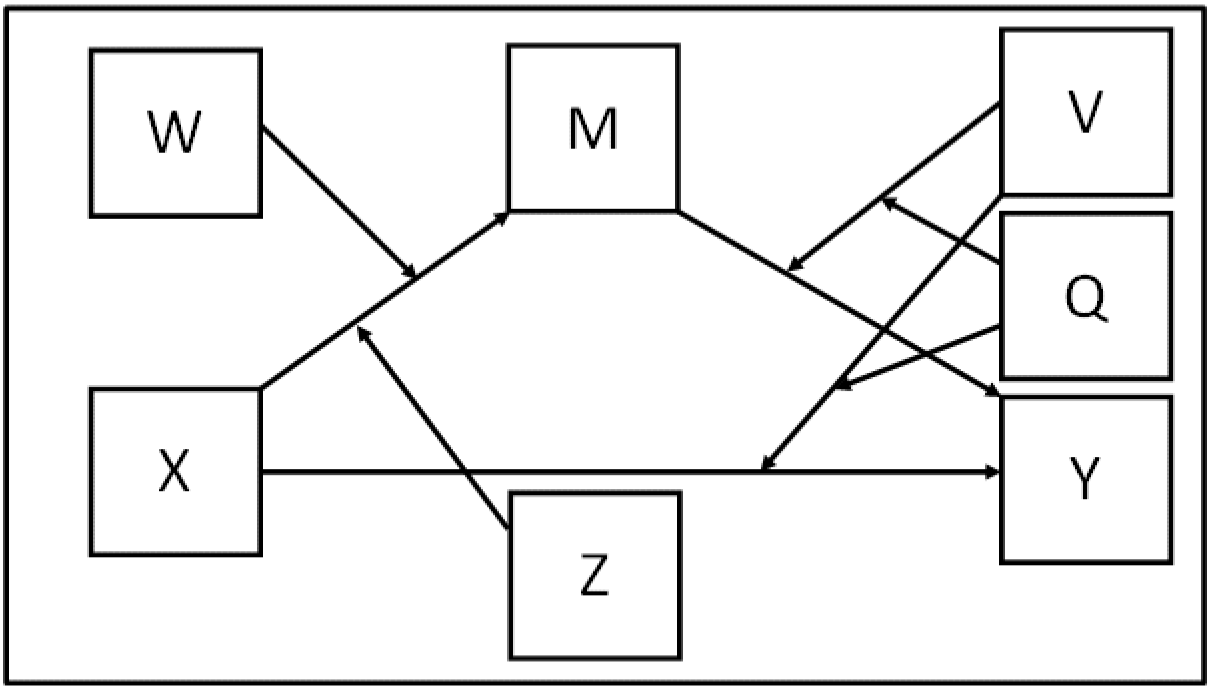
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

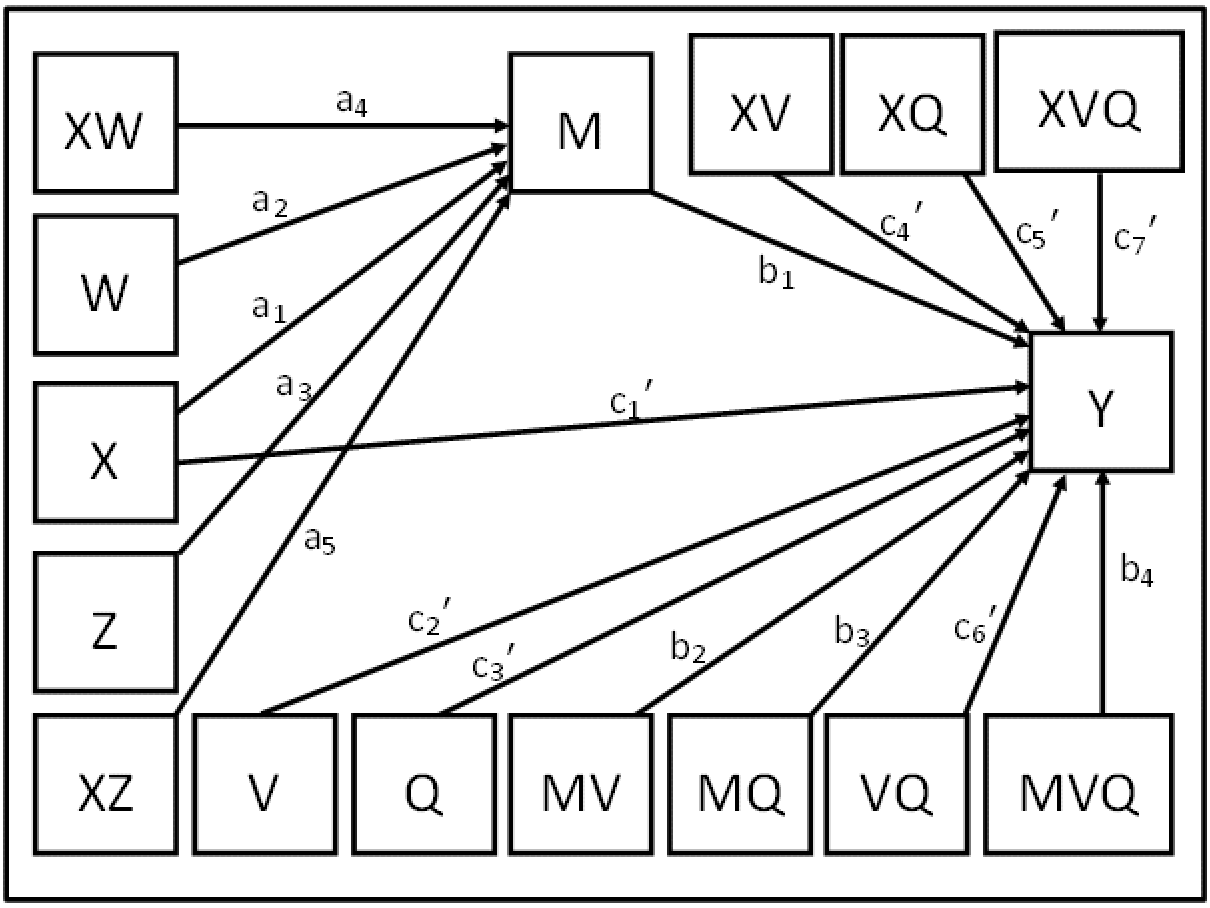
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ)V + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ)Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ)VQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a0b2V + a1b2XV + a2b2WV + a3b2ZV + a4b2XWV + a5b2XZV + a0b3Q + a1b3XQ + a2b3WQ + a3b3ZQ + a4b3XWQ + a5b3XZQ + a0b4VQ + a1b4XVQ + a2b4WVQ + a3b4ZVQ + a4b4XWVQ + a5b4XZVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a0b2V + a2b2WV + a3b2ZV + a0b3Q + a2b3WQ + a3b3ZQ + a0b4VQ + a2b4WVQ + a3b4ZVQ + c2'V + c3'Q + c6'VQ) + (a1b1 + a4b1W + a5b1Z + a1b2V + a4b2WV + a5b2ZV + a1b3Q + a4b3WQ + a5b3ZQ + a1b4VQ + a4b4WVQ + a5b4ZVQ + c1' + c4'V + c5'Q + c7'VQ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a1b2V + a4b2WV + a5b2ZV + a1b3Q + a4b3WQ + a5b3ZQ + a1b4VQ + a4b4WVQ + a5b4ZVQ = (a1 + a4W + a5Z) (b1 + b2V + b3Q + b4VQ)

One direct effect of X on Y, conditional on V, Q:

c1' + c4'V + c5'Q + c7'VQ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ XV XQ VQ MV MQ XVQ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   XQ = X*Q;
   XV = X*V;
   VQ = V*Q;
```

```
   MVQ = M*V*Q;
   XVQ = X*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MV (b2);
   Y ON MQ (b3);
   Y ON MVQ (b4);

   Y ON X (cdash1);
   Y ON V (cdash2);
   Y ON Q (cdash3);
   Y ON XV (cdash4);
   Y ON XQ (cdash5);
   Y ON VQ (cdash6);
   Y ON XVQ (cdash7);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
```

```
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    DLOV_LOQ DMEV_LOQ DHIV_LOQ DLOV_MEQ DMEV_MEQ DHIV_MEQ
    DLOV_HIQ DMEV_HIQ DHIV_HIQ
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values
```

```
    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_V +
     a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*LOW_Z*LOW_Q + a1*b4*LOW_V*LOW_Q
+
     a4*b4*LOW_W*LOW_V*LOW_Q + a5*b4*LOW_Z*LOW_V*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*LOW_V +
     a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*LOW_Z*LOW_Q + a1*b4*LOW_V*LOW_Q
+
     a4*b4*MED_W*LOW_V*LOW_Q + a5*b4*LOW_Z*LOW_V*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*LOW_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*HIGH_W*LOW_V*LOW_Q + a5*b4*LOW_Z*LOW_V*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_V +
     a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*MED_Z*LOW_Q + a1*b4*LOW_V*LOW_Q
+
     a4*b4*LOW_W*LOW_V*LOW_Q + a5*b4*MED_Z*LOW_V*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*LOW_V +
     a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*MED_Z*LOW_Q + a1*b4*LOW_V*LOW_Q
+
     a4*b4*MED_W*LOW_V*LOW_Q + a5*b4*MED_Z*LOW_V*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*MED_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*HIGH_W*LOW_V*LOW_Q + a5*b4*MED_Z*LOW_V*LOW_Q;

    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*LOW_W*LOW_V*LOW_Q + a5*b4*HIGH_Z*LOW_V*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*MED_W*LOW_V*LOW_Q + a5*b4*HIGH_Z*LOW_V*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
```

```
     a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*HIGH_W*LOW_V*LOW_Q + a5*b4*HIGH_Z*LOW_V*LOW_Q;

    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*MED_V +
     a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*LOW_Z*LOW_Q + a1*b4*MED_V*LOW_Q
+
     a4*b4*LOW_W*MED_V*LOW_Q + a5*b4*LOW_Z*MED_V*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_V +
     a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*LOW_Z*LOW_Q + a1*b4*MED_V*LOW_Q
+
     a4*b4*MED_W*MED_V*LOW_Q + a5*b4*LOW_Z*MED_V*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*LOW_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*HIGH_W*MED_V*LOW_Q + a5*b4*LOW_Z*MED_V*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*MED_V +
     a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*MED_Z*LOW_Q + a1*b4*MED_V*LOW_Q
+
     a4*b4*LOW_W*MED_V*LOW_Q + a5*b4*MED_Z*MED_V*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_V +
     a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*MED_Z*LOW_Q + a1*b4*MED_V*LOW_Q
+
     a4*b4*MED_W*MED_V*LOW_Q + a5*b4*MED_Z*MED_V*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*MED_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*HIGH_W*MED_V*LOW_Q + a5*b4*MED_Z*MED_V*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*LOW_W*MED_V*LOW_Q + a5*b4*HIGH_Z*MED_V*LOW_Q;
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q +
```

```
a1*b4*MED_V*LOW_Q +
     a4*b4*MED_W*MED_V*LOW_Q + a5*b4*HIGH_Z*MED_V*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*HIGH_W*MED_V*LOW_Q + a5*b4*HIGH_Z*MED_V*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*LOW_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*LOW_W*HIGH_V*LOW_Q + a5*b4*LOW_Z*HIGH_V*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*LOW_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*MED_W*HIGH_V*LOW_Q + a5*b4*LOW_Z*HIGH_V*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*LOW_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*HIGH_W*HIGH_V*LOW_Q + a5*b4*LOW_Z*HIGH_V*LOW_Q;

    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*MED_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*LOW_W*HIGH_V*LOW_Q + a5*b4*MED_Z*HIGH_V*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*MED_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*MED_W*HIGH_V*LOW_Q + a5*b4*MED_Z*HIGH_V*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*HIGH_W*LOW_Q + a5*b3*MED_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*HIGH_W*HIGH_V*LOW_Q + a5*b4*MED_Z*HIGH_V*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
```

```
     a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*LOW_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*LOW_W*HIGH_V*LOW_Q + a5*b4*HIGH_Z*HIGH_V*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
     a4*b3*MED_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*MED_W*HIGH_V*LOW_Q + a5*b4*HIGH_Z*HIGH_V*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*LOW_Q
+
     a4*b3*HIGH_W*LOW_Q + a5*b3*HIGH_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*HIGH_W*HIGH_V*LOW_Q + a5*b4*HIGH_Z*HIGH_V*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_V +
     a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*MED_Q +
     a4*b3*LOW_W*MED_Q + a5*b3*LOW_Z*MED_Q + a1*b4*LOW_V*MED_Q
+
     a4*b4*LOW_W*LOW_V*MED_Q + a5*b4*LOW_Z*LOW_V*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*LOW_V +
     a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*MED_Q +
     a4*b3*MED_W*MED_Q + a5*b3*LOW_Z*MED_Q + a1*b4*LOW_V*MED_Q
+
     a4*b4*MED_W*LOW_V*MED_Q + a5*b4*LOW_Z*LOW_V*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*MED_Q +
     a4*b3*HIGH_W*MED_Q + a5*b3*LOW_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*HIGH_W*LOW_V*MED_Q + a5*b4*LOW_Z*LOW_V*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_V +
     a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*MED_Q +
     a4*b3*LOW_W*MED_Q + a5*b3*MED_Z*MED_Q + a1*b4*LOW_V*MED_Q
+
     a4*b4*LOW_W*LOW_V*MED_Q + a5*b4*MED_Z*LOW_V*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*LOW_V +
     a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*MED_Q +
     a4*b3*MED_W*MED_Q + a5*b3*MED_Z*MED_Q + a1*b4*LOW_V*MED_Q
+
     a4*b4*MED_W*LOW_V*MED_Q + a5*b4*MED_Z*LOW_V*MED_Q;
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*MED_Q +
```

```
     a4*b3*HIGH_W*MED_Q + a5*b3*MED_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*HIGH_W*LOW_V*MED_Q + a5*b4*MED_Z*LOW_V*MED_Q;
    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*MED_Q +
     a4*b3*LOW_W*MED_Q + a5*b3*HIGH_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*LOW_W*LOW_V*MED_Q + a5*b4*HIGH_Z*LOW_V*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*MED_Q +
     a4*b3*MED_W*MED_Q + a5*b3*HIGH_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*MED_W*LOW_V*MED_Q + a5*b4*HIGH_Z*LOW_V*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*MED_Q +
     a4*b3*HIGH_W*MED_Q + a5*b3*HIGH_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*HIGH_W*LOW_V*MED_Q + a5*b4*HIGH_Z*LOW_V*MED_Q;
    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*MED_V +
     a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*MED_Q +
     a4*b3*LOW_W*MED_Q + a5*b3*LOW_Z*MED_Q + a1*b4*MED_V*MED_Q
+
     a4*b4*LOW_W*MED_V*MED_Q + a5*b4*LOW_Z*MED_V*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_V +
     a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*MED_Q +
     a4*b3*MED_W*MED_Q + a5*b3*LOW_Z*MED_Q + a1*b4*MED_V*MED_Q
+
     a4*b4*MED_W*MED_V*MED_Q + a5*b4*LOW_Z*MED_V*MED_Q;
    IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*MED_Q +
     a4*b3*HIGH_W*MED_Q + a5*b3*LOW_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*HIGH_W*MED_V*MED_Q + a5*b4*LOW_Z*MED_V*MED_Q;
    ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*MED_V +
     a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*MED_Q +
     a4*b3*LOW_W*MED_Q + a5*b3*MED_Z*MED_Q + a1*b4*MED_V*MED_Q
+
     a4*b4*LOW_W*MED_V*MED_Q + a5*b4*MED_Z*MED_V*MED_Q;
    IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_V +
     a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*MED_Q +
     a4*b3*MED_W*MED_Q + a5*b3*MED_Z*MED_Q + a1*b4*MED_V*MED_Q
+
```

```
     a4*b4*MED_W*MED_V*MED_Q + a5*b4*MED_Z*MED_V*MED_Q;
    IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*MED_Q +
     a4*b3*HIGH_W*MED_Q + a5*b3*MED_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*HIGH_W*MED_V*MED_Q + a5*b4*MED_Z*MED_V*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*MED_Q +
     a4*b3*LOW_W*MED_Q + a5*b3*HIGH_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*LOW_W*MED_V*MED_Q + a5*b4*HIGH_Z*MED_V*MED_Q;
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*MED_Q +
     a4*b3*MED_W*MED_Q + a5*b3*HIGH_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*MED_W*MED_V*MED_Q + a5*b4*HIGH_Z*MED_V*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*MED_Q +
     a4*b3*HIGH_W*MED_Q + a5*b3*HIGH_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*HIGH_W*MED_V*MED_Q + a5*b4*HIGH_Z*MED_V*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*MED_Q +
     a4*b3*LOW_W*MED_Q + a5*b3*LOW_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*LOW_W*HIGH_V*MED_Q + a5*b4*LOW_Z*HIGH_V*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*MED_Q +
     a4*b3*MED_W*MED_Q + a5*b3*LOW_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*MED_W*HIGH_V*MED_Q + a5*b4*LOW_Z*HIGH_V*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*MED_Q +
     a4*b3*HIGH_W*MED_Q + a5*b3*LOW_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*HIGH_W*HIGH_V*MED_Q + a5*b4*LOW_Z*HIGH_V*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*MED_Q +
```

```
     a4*b3*LOW_W*MED_Q + a5*b3*MED_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*LOW_W*HIGH_V*MED_Q + a5*b4*MED_Z*HIGH_V*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*MED_Q +
     a4*b3*MED_W*MED_Q + a5*b3*MED_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*MED_W*HIGH_V*MED_Q + a5*b4*MED_Z*HIGH_V*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*MED_Q +
     a4*b3*HIGH_W*MED_Q + a5*b3*MED_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*HIGH_W*HIGH_V*MED_Q + a5*b4*MED_Z*HIGH_V*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*MED_Q +
     a4*b3*LOW_W*MED_Q + a5*b3*HIGH_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*LOW_W*HIGH_V*MED_Q + a5*b4*HIGH_Z*HIGH_V*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*MED_Q +
     a4*b3*MED_W*MED_Q + a5*b3*HIGH_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*MED_W*HIGH_V*MED_Q + a5*b4*HIGH_Z*HIGH_V*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*MED_Q
+
     a4*b3*HIGH_W*MED_Q + a5*b3*HIGH_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*HIGH_W*HIGH_V*MED_Q + a5*b4*HIGH_Z*HIGH_V*MED_Q;

    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_V +
     a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*HIGH_Q +
     a4*b3*LOW_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*LOW_W*LOW_V*HIGH_Q + a5*b4*LOW_Z*LOW_V*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*LOW_V +
     a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*HIGH_Q +
     a4*b3*MED_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*MED_W*LOW_V*HIGH_Q + a5*b4*LOW_Z*LOW_V*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*LOW_V
+
```

```
     a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V + a1*b3*HIGH_Q +
     a4*b3*HIGH_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*HIGH_W*LOW_V*HIGH_Q + a5*b4*LOW_Z*LOW_V*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_V +
     a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*HIGH_Q +
     a4*b3*LOW_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*LOW_W*LOW_V*HIGH_Q + a5*b4*MED_Z*LOW_V*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*LOW_V +
     a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*HIGH_Q +
     a4*b3*MED_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*MED_W*LOW_V*HIGH_Q + a5*b4*MED_Z*LOW_V*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V + a1*b3*HIGH_Q +
     a4*b3*HIGH_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*HIGH_W*LOW_V*HIGH_Q + a5*b4*MED_Z*LOW_V*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
     a4*b3*LOW_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*LOW_W*LOW_V*HIGH_Q + a5*b4*HIGH_Z*LOW_V*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
     a4*b3*MED_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*MED_W*LOW_V*HIGH_Q + a5*b4*HIGH_Z*LOW_V*HIGH_Q;
    IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*LOW_V
+
     a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
     a4*b3*HIGH_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*HIGH_W*LOW_V*HIGH_Q + a5*b4*HIGH_Z*LOW_V*HIGH_Q;

    ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*MED_V +
     a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*HIGH_Q +
     a4*b3*LOW_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*LOW_W*MED_V*HIGH_Q + a5*b4*LOW_Z*MED_V*HIGH_Q;
    IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_V +
     a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*HIGH_Q +
     a4*b3*MED_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q +
```

```
a1*b4*MED_V*HIGH_Q +
     a4*b4*MED_W*MED_V*HIGH_Q + a5*b4*LOW_Z*MED_V*HIGH_Q;
    IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V + a1*b3*HIGH_Q +
     a4*b3*HIGH_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*HIGH_W*MED_V*HIGH_Q + a5*b4*LOW_Z*MED_V*HIGH_Q;

    ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*MED_V +
     a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*HIGH_Q +
     a4*b3*LOW_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*LOW_W*MED_V*HIGH_Q + a5*b4*MED_Z*MED_V*HIGH_Q;
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_V +
     a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*HIGH_Q +
     a4*b3*MED_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*MED_W*MED_V*HIGH_Q + a5*b4*MED_Z*MED_V*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V + a1*b3*HIGH_Q +
     a4*b3*HIGH_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*HIGH_W*MED_V*HIGH_Q + a5*b4*MED_Z*MED_V*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*HIGH_Q +
     a4*b3*LOW_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*LOW_W*MED_V*HIGH_Q + a5*b4*HIGH_Z*MED_V*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*HIGH_Q +
     a4*b3*MED_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*MED_W*MED_V*HIGH_Q + a5*b4*HIGH_Z*MED_V*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*MED_V
+
     a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V + a1*b3*HIGH_Q +
     a4*b3*HIGH_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*HIGH_W*MED_V*HIGH_Q + a5*b4*HIGH_Z*MED_V*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
     a4*b3*LOW_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q +
```

```
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*LOW_W*HIGH_V*HIGH_Q + a5*b4*LOW_Z*HIGH_V*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
     a4*b3*MED_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*MED_W*HIGH_V*HIGH_Q + a5*b4*LOW_Z*HIGH_V*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V + a1*b3*HIGH_Q
+
     a4*b3*HIGH_W*HIGH_Q + a5*b3*LOW_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*HIGH_W*HIGH_V*HIGH_Q + a5*b4*LOW_Z*HIGH_V*HIGH_Q;

    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*HIGH_Q +
     a4*b3*LOW_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*LOW_W*HIGH_V*HIGH_Q + a5*b4*MED_Z*HIGH_V*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*HIGH_Q +
     a4*b3*MED_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*MED_W*HIGH_V*HIGH_Q + a5*b4*MED_Z*HIGH_V*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V + a1*b3*HIGH_Q
+
     a4*b3*HIGH_W*HIGH_Q + a5*b3*MED_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*HIGH_W*HIGH_V*HIGH_Q + a5*b4*MED_Z*HIGH_V*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*HIGH_Q
+
     a4*b3*LOW_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*LOW_W*HIGH_V*HIGH_Q + a5*b4*HIGH_Z*HIGH_V*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*HIGH_Q
+
     a4*b3*MED_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q +
```

```
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*MED_W*HIGH_V*HIGH_Q + a5*b4*HIGH_Z*HIGH_V*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z + a1*b2*HIGH_V
+
     a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V + a1*b3*HIGH_Q
+
     a4*b3*HIGH_W*HIGH_Q + a5*b3*HIGH_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*HIGH_W*HIGH_V*HIGH_Q + a5*b4*HIGH_Z*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOV_LOQ = cdash1 + cdash4*LOW_V + cdash5*LOW_Q +
cdash7*LOW_V*LOW_Q;
    DMEV_LOQ = cdash1 + cdash4*MED_W + cdash5*LOW_Q +
cdash7*MED_W*LOW_Q;
    DHIV_LOQ = cdash1 + cdash4*HIGH_V + cdash5*LOW_Q +
cdash7*HIGH_V*LOW_Q;

    DLOV_MEQ = cdash1 + cdash4*LOW_V + cdash5*MED_Q +
cdash7*LOW_V*MED_Q;
    DMEV_MEQ = cdash1 + cdash4*MED_W + cdash5*MED_Q +
cdash7*MED_W*MED_Q;
    DHIV_MEQ = cdash1 + cdash4*HIGH_V + cdash5*MED_Q +
cdash7*HIGH_V*MED_Q;

    DLOV_HIQ = cdash1 + cdash4*LOW_V + cdash5*HIGH_Q +
cdash7*LOW_V*HIGH_Q;
    DMEV_HIQ = cdash1 + cdash4*MED_W + cdash5*HIGH_Q +
cdash7*MED_W*HIGH_Q;
    DHIV_HIQ = cdash1 + cdash4*HIGH_V + cdash5*HIGH_Q +
cdash7*HIGH_V*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + DLOV_LOQ;
    TMLLL = IMLLL + DLOV_LOQ;
    THLLL = IHLLL + DLOV_LOQ;

    TLMLL = ILMLL + DLOV_LOQ;
    TMMLL = IMMLL + DLOV_LOQ;
    THMLL = IHMLL + DLOV_LOQ;

    TLHLL = ILHLL + DLOV_LOQ;
    TMHLL = IMHLL + DLOV_LOQ;
    THHLL = IHHLL + DLOV_LOQ;
```

```
TLLML = ILLML + DMEV_LOQ;
TMLML = IMLML + DMEV_LOQ;
THLML = IHLML + DMEV_LOQ;

TLMML = ILMML + DMEV_LOQ;
TMMML = IMMML + DMEV_LOQ;
THMML = IHMML + DMEV_LOQ;

TLHML = ILHML + DMEV_LOQ;
TMHML = IMHML + DMEV_LOQ;
THHML = IHHML + DMEV_LOQ;

TLLHL = ILLHL + DHIV_LOQ;
TMLHL = IMLHL + DHIV_LOQ;
THLHL = IHLHL + DHIV_LOQ;

TLMHL = ILMHL + DHIV_LOQ;
TMMHL = IMMHL + DHIV_LOQ;
THMHL = IHMHL + DHIV_LOQ;

TLHHL = ILHHL + DHIV_LOQ;
TMHHL = IMHHL + DHIV_LOQ;
THHHL = IHHHL + DHIV_LOQ;

TLLLM = ILLLM + DLOV_MEQ;
TMLLM = IMLLM + DLOV_MEQ;
THLLM = IHLLM + DLOV_MEQ;

TLMLM = ILMLM + DLOV_MEQ;
TMMLM = IMMLM + DLOV_MEQ;
THMLM = IHMLM + DLOV_MEQ;

TLHLM = ILHLM + DLOV_MEQ;
TMHLM = IMHLM + DLOV_MEQ;
THHLM = IHHLM + DLOV_MEQ;

TLLMM = ILLMM + DMEV_MEQ;
TMLMM = IMLMM + DMEV_MEQ;
THLMM = IHLMM + DMEV_MEQ;

TLMMM = ILMMM + DMEV_MEQ;
TMMMM = IMMMM + DMEV_MEQ;
THMMM = IHMMM + DMEV_MEQ;

TLHMM = ILHMM + DMEV_MEQ;
TMHMM = IMHMM + DMEV_MEQ;
THHMM = IHHMM + DMEV_MEQ;

TLLHM = ILLHM + DHIV_MEQ;
TMLHM = IMLHM + DHIV_MEQ;
THLHM = IHLHM + DHIV_MEQ;
```

```
    TLMHM = ILMHM + DHIV_MEQ;
    TMMHM = IMMHM + DHIV_MEQ;
    THMHM = IHMHM + DHIV_MEQ;

    TLHHM = ILHHM + DHIV_MEQ;
    TMHHM = IMHHM + DHIV_MEQ;
    THHHM = IHHHM + DHIV_MEQ;

    TLLLH = ILLLH + DLOV_HIQ;
    TMLLH = IMLLH + DLOV_HIQ;
    THLLH = IHLLH + DLOV_HIQ;

    TLMLH = ILMLH + DLOV_HIQ;
    TMMLH = IMMLH + DLOV_HIQ;
    THMLH = IHMLH + DLOV_HIQ;

    TLHLH = ILHLH + DLOV_HIQ;
    TMHLH = IMHLH + DLOV_HIQ;
    THHLH = IHHLH + DLOV_HIQ;

    TLLMH = ILLMH + DMEV_HIQ;
    TMLMH = IMLMH + DMEV_HIQ;
    THLMH = IHLMH + DMEV_HIQ;

    TLMMH = ILMMH + DMEV_HIQ;
    TMMMH = IMMMH + DMEV_HIQ;
    THMMH = IHMMH + DMEV_HIQ;

    TLHMH = ILHMH + DMEV_HIQ;
    TMHMH = IMHMH + DMEV_HIQ;
    THHMH = IHHMH + DMEV_HIQ;

    TLLHH = ILLHH + DHIV_HIQ;
    TMLHH = IMLHH + DHIV_HIQ;
    THLHH = IHLHH + DHIV_HIQ;

    TLMHH = ILMHH + DHIV_HIQ;
    TMMHH = IMMHH + DHIV_HIQ;
    THMHH = IHMHH + DHIV_HIQ;

    TLHHH = ILHHH + DHIV_HIQ;
    TMHHH = IMHHH + DHIV_HIQ;
    THHHH = IHHHH + DHIV_HIQ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis
```

```
    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;

    PLMLL = ILMLL*XVAL;
    PMMLL = IMMLL*XVAL;
    PHMLL = IHMLL*XVAL;

    PLHLL = ILHLL*XVAL;
    PMHLL = IMHLL*XVAL;
    PHHLL = IHHLL*XVAL;

    PLLML = ILLML*XVAL;
    PMLML = IMLML*XVAL;
    PHLML = IHLML*XVAL;

    PLMML = ILMML*XVAL;
    PMMML = IMMML*XVAL;
    PHMML = IHMML*XVAL;

    PLHML = ILHML*XVAL;
    PMHML = IMHML*XVAL;
    PHHML = IHHML*XVAL;

    PLLHL = ILLHL*XVAL;
    PMLHL = IMLHL*XVAL;
    PHLHL = IHLHL*XVAL;

    PLMHL = ILMHL*XVAL;
    PMMHL = IMMHL*XVAL;
    PHMHL = IHMHL*XVAL;

    PLHHL = ILHHL*XVAL;
    PMHHL = IMHHL*XVAL;
    PHHHL = IHHHL*XVAL;

    PLLLM = ILLLM*XVAL;
    PMLLM = IMLLM*XVAL;
    PHLLM = IHLLM*XVAL;
```

```
PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;

PLHLH = ILHLH*XVAL;
PMHLH = IMHLH*XVAL;
PHHLH = IHHLH*XVAL;

PLLMH = ILLMH*XVAL;
PMLMH = IMLMH*XVAL;
PHLMH = IHLMH*XVAL;

PLMMH = ILMMH*XVAL;
PMMMH = IMMMH*XVAL;
PHMMH = IHMMH*XVAL;
```

```
     PLHMH = ILHMH*XVAL;
     PMHMH = IMHMH*XVAL;
     PHHMH = IHHMH*XVAL;

     PLLHH = ILLHH*XVAL;
     PMLHH = IMLHH*XVAL;
     PHLHH = IHLHH*XVAL;

     PLMHH = ILMHH*XVAL;
     PMMHH = IMMHH*XVAL;
     PHMHH = IHMHH*XVAL;

     PLHHH = ILHHH*XVAL;
     PMHHH = IMHHH*XVAL;
     PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 55: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path with all 2-way and 3-way interactions, with the other 2 moderating the Mediator-DV path with all 2-way and 3-way interactions

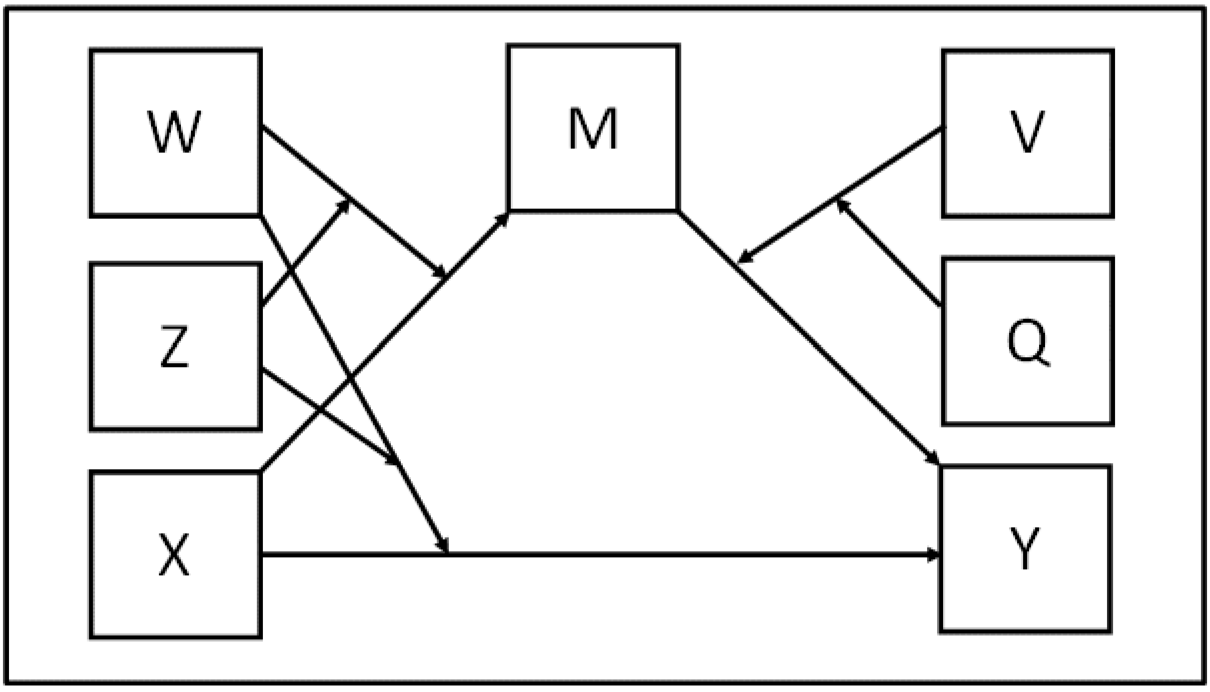
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

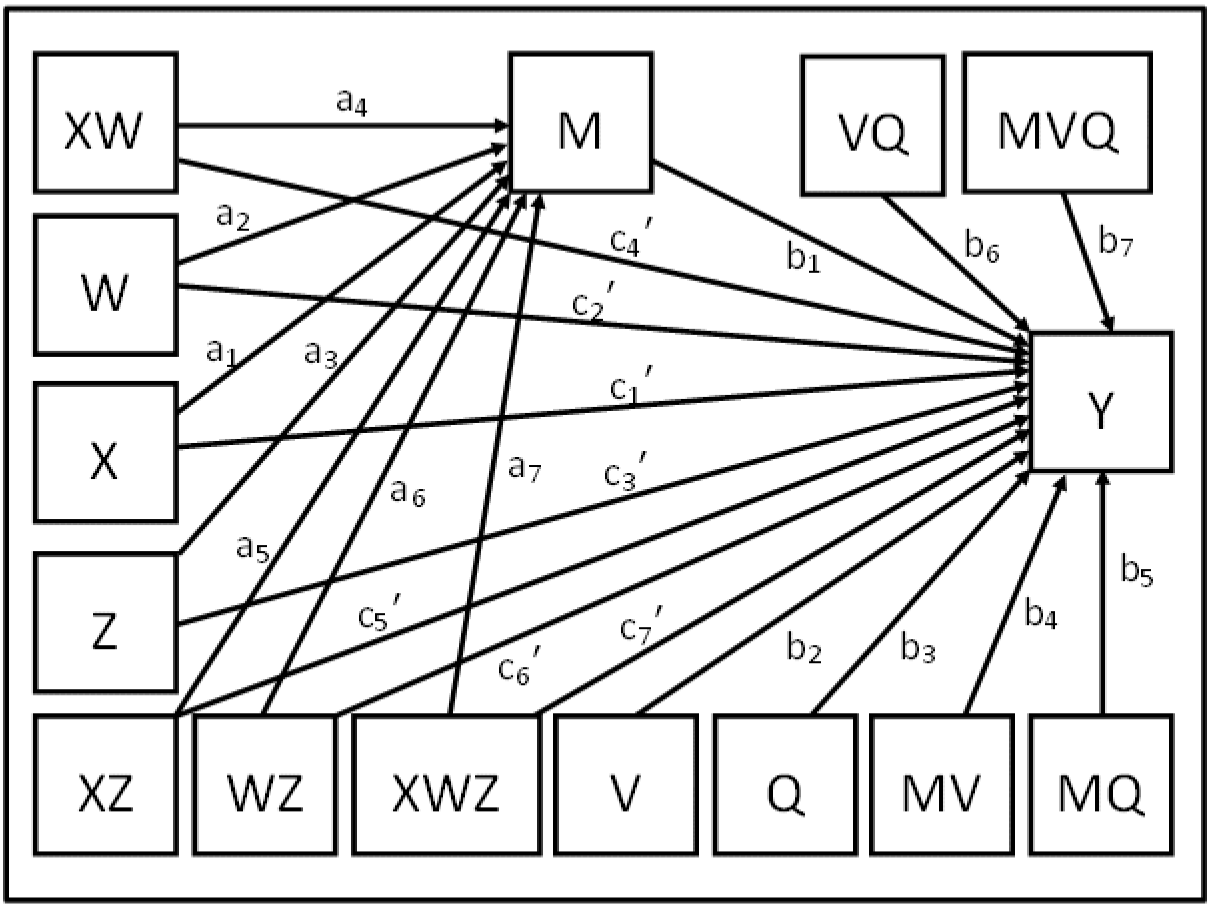
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3Q + b4MV + b5MQ + b6VQ + b7MVQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2V + b3Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + b5(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)Q + b6VQ + b7(a0 + a1X + a2W +

a3Z + a4XW + a5XZ + a6WZ + a7XWZ)VQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + b2V + b3Q + a0b4V + a1b4XV + a2b4WV + a3b4ZV + a4b4XWV + a5b4XZV + a6b4WZV + a7b4XWZV + a0b5Q + a1b5XQ + a2b5WQ + a3b5ZQ + a4b5XWQ + a5b5XZQ + a6b5WZQ + a7b5XWZQ + b6VQ + a0b7VQ + a1b7XVQ + a2b7WVQ + a3b7ZVQ + a4b7XWVQ + a5b7XZVQ + a6b7WZVQ + a7b7XWZVQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + b2V + b3Q + a0b4V + a2b4WV + a3b4ZV + a6b4WZV + a0b5Q + a2b5WQ + a3b5ZQ + a6b5WZQ + b6VQ + a0b7VQ + a2b7WVQ + a3b7ZVQ + a6b7WZVQ + c2'W + c3'Z + c6'WZ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4V + a4b4WV + a5b4ZV + a7b4WZV + a1b5Q + a4b5WQ + a5b5ZQ + a7b5WZQ + a1b7VQ+ a4b7WVQ + a5b7ZVQ + a7b7WZVQ + c1' + c4'W + c5'Z + c7'WZ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4V + a4b4WV + a5b4ZV + a7b4WZV + a1b5Q + a4b5WQ + a5b5ZQ + a7b5WZQ + a1b7VQ+ a4b7WVQ + a5b7ZVQ + a7b7WZVQ = (a1 + a4W + a5Z + a7WZ)(b1 + b4V + b5Q + b7VQ)

One direct effect of X on Y, conditional on W, Z:

c1' + c4'W + c5'Z + c7'WZ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ WZ VQ MV MQ XWZ MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
   MV = M*V;
```

```
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   VQ = V*Q;
   MVQ = M*V*Q;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON Q (b3);
   Y ON MV (b4);
   Y ON MQ (b5);
   Y ON VQ (b6);
   Y ON MVQ (b7);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON Z (cdash3);
   Y ON XW (cdash4);
   Y ON XZ (cdash5);
   Y ON WZ (cdash6);
   Y ON XWZ (cdash7);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
```

```
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V
```

```
    LOW_Q = #LOWQ;    ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;    ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;    ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q +
     a7*b7*LOW_W*LOW_Z*LOW_V*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q +
     a7*b7*MED_W*LOW_Z*LOW_V*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*LOW_Z*LOW_V*LOW_Q +
     a7*b7*HIGH_W*LOW_Z*LOW_V*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q +
     a7*b7*LOW_W*MED_Z*LOW_V*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
```

```
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q +
     a7*b7*MED_W*MED_Z*LOW_V*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*MED_Z*LOW_V*LOW_Q +
     a7*b7*HIGH_W*MED_Z*LOW_V*LOW_Q;

    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*LOW_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q +
     a7*b7*LOW_W*HIGH_Z*LOW_V*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*MED_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q +
     a7*b7*MED_W*HIGH_Z*LOW_V*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q +
a1*b7*LOW_V*LOW_Q +
     a4*b7*HIGH_W*LOW_V*LOW_Q + a5*b7*HIGH_Z*LOW_V*LOW_Q +
     a7*b7*HIGH_W*HIGH_Z*LOW_V*LOW_Q;

    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
```

```
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q +
     a7*b7*LOW_W*LOW_Z*MED_V*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q +
     a7*b7*MED_W*LOW_Z*MED_V*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*LOW_Z*MED_V*LOW_Q +
     a7*b7*HIGH_W*LOW_Z*MED_V*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*LOW_Q + a4*b5*LOW_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q +
     a7*b7*LOW_W*MED_Z*MED_V*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*LOW_Q + a4*b5*MED_W*LOW_Q
+
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q +
     a7*b7*MED_W*MED_Z*MED_V*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
```

```
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*MED_Z*MED_V*LOW_Q +
     a7*b7*HIGH_W*MED_Z*MED_V*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*LOW_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q +
     a7*b7*LOW_W*HIGH_Z*MED_V*LOW_Q;
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*MED_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q +
     a7*b7*MED_W*HIGH_Z*MED_V*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q +
a1*b7*MED_V*LOW_Q +
     a4*b7*HIGH_W*MED_V*LOW_Q + a5*b7*HIGH_Z*MED_V*LOW_Q +
     a7*b7*HIGH_W*HIGH_Z*MED_V*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*LOW_W*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q +
     a7*b7*LOW_W*LOW_Z*HIGH_V*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*MED_W*LOW_Z*LOW_Q +
```

```
a1*b7*HIGH_V*LOW_Q +
     a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q +
     a7*b7*MED_W*LOW_Z*HIGH_V*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*LOW_Z*LOW_Q + a7*b5*HIGH_W*LOW_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*LOW_Z*HIGH_V*LOW_Q +
     a7*b7*HIGH_W*LOW_Z*HIGH_V*LOW_Q;

    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*LOW_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*LOW_W*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q +
     a7*b7*LOW_W*MED_Z*HIGH_V*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*MED_W*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q +
     a7*b7*MED_W*MED_Z*HIGH_V*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*MED_Z*LOW_Q + a7*b5*HIGH_W*MED_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*MED_Z*HIGH_V*LOW_Q +
     a7*b7*HIGH_W*MED_Z*HIGH_V*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
```

```
a4*b5*LOW_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*LOW_W*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*LOW_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q +
     a7*b7*LOW_W*HIGH_Z*HIGH_V*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*MED_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*MED_W*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*MED_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q +
     a7*b7*MED_W*HIGH_Z*HIGH_V*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*LOW_Q +
a4*b5*HIGH_W*LOW_Q +
     a5*b5*HIGH_Z*LOW_Q + a7*b5*HIGH_W*HIGH_Z*LOW_Q +
a1*b7*HIGH_V*LOW_Q +
     a4*b7*HIGH_W*HIGH_V*LOW_Q + a5*b7*HIGH_Z*HIGH_V*LOW_Q +
     a7*b7*HIGH_W*HIGH_Z*HIGH_V*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q +
     a7*b7*LOW_W*LOW_Z*LOW_V*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q +
     a7*b7*MED_W*LOW_Z*LOW_V*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*MED_Q +
```

```
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*LOW_Z*LOW_V*MED_Q +
     a7*b7*HIGH_W*LOW_Z*LOW_V*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q +
     a7*b7*LOW_W*MED_Z*LOW_V*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q +
     a7*b7*MED_W*MED_Z*LOW_V*MED_Q;
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*MED_Z*LOW_V*MED_Q +
     a7*b7*HIGH_W*MED_Z*LOW_V*MED_Q;

    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*LOW_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q +
     a7*b7*LOW_W*HIGH_Z*LOW_V*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
```

```
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*MED_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q +
     a7*b7*MED_W*HIGH_Z*LOW_V*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q +
a1*b7*LOW_V*MED_Q +
     a4*b7*HIGH_W*LOW_V*MED_Q + a5*b7*HIGH_Z*LOW_V*MED_Q +
     a7*b7*HIGH_W*HIGH_Z*LOW_V*MED_Q;

    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q +
     a7*b7*LOW_W*LOW_Z*MED_V*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q +
     a7*b7*MED_W*LOW_Z*MED_V*MED_Q;
    IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*LOW_Z*MED_V*MED_Q +
     a7*b7*HIGH_W*LOW_Z*MED_V*MED_Q;

    ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*MED_Q + a4*b5*LOW_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q +
```

```
a1*b7*MED_V*MED_Q +
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q +
     a7*b7*LOW_W*MED_Z*MED_V*MED_Q;
    IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*MED_Q + a4*b5*MED_W*MED_Q
+
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q +
     a7*b7*MED_W*MED_Z*MED_V*MED_Q;
    IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*MED_Z*MED_V*MED_Q +
     a7*b7*HIGH_W*MED_Z*MED_V*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*LOW_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q +
     a7*b7*LOW_W*HIGH_Z*MED_V*MED_Q;
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
     a4*b7*MED_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q +
     a7*b7*MED_W*HIGH_Z*MED_V*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q +
a1*b7*MED_V*MED_Q +
```

```
     a4*b7*HIGH_W*MED_V*MED_Q + a5*b7*HIGH_Z*MED_V*MED_Q +
     a7*b7*HIGH_W*HIGH_Z*MED_V*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*LOW_W*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q +
     a7*b7*LOW_W*LOW_Z*HIGH_V*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*MED_W*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q +
     a7*b7*MED_W*LOW_Z*HIGH_V*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*LOW_Z*MED_Q + a7*b5*HIGH_W*LOW_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*LOW_Z*HIGH_V*MED_Q +
     a7*b7*HIGH_W*LOW_Z*HIGH_V*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*LOW_W*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q +
     a7*b7*LOW_W*MED_Z*HIGH_V*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*MED_W*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
```

```
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q +
     a7*b7*MED_W*MED_Z*HIGH_V*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*MED_Z*MED_Q + a7*b5*HIGH_W*MED_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*MED_Z*HIGH_V*MED_Q +
     a7*b7*HIGH_W*MED_Z*HIGH_V*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*LOW_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*LOW_W*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*LOW_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q +
     a7*b7*LOW_W*HIGH_Z*HIGH_V*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*MED_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*MED_W*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*MED_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q +
     a7*b7*MED_W*HIGH_Z*HIGH_V*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*MED_Q +
a4*b5*HIGH_W*MED_Q +
     a5*b5*HIGH_Z*MED_Q + a7*b5*HIGH_W*HIGH_Z*MED_Q +
a1*b7*HIGH_V*MED_Q +
     a4*b7*HIGH_W*HIGH_V*MED_Q + a5*b7*HIGH_Z*HIGH_V*MED_Q +
     a7*b7*HIGH_W*HIGH_Z*HIGH_V*MED_Q;

    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*LOW_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
```

```
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q +
     a7*b7*LOW_W*LOW_Z*LOW_V*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*MED_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q +
     a7*b7*MED_W*LOW_Z*LOW_V*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*LOW_Z*LOW_V +
     a7*b4*HIGH_W*LOW_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*LOW_Z*LOW_V*HIGH_Q +
     a7*b7*HIGH_W*LOW_Z*LOW_V*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*LOW_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q +
     a7*b7*LOW_W*MED_Z*LOW_V*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*MED_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q +
     a7*b7*MED_W*MED_Z*LOW_V*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*MED_Z*LOW_V +
     a7*b4*HIGH_W*MED_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q +
```

```
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*MED_Z*LOW_V*HIGH_Q +
     a7*b7*HIGH_W*MED_Z*LOW_V*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*LOW_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*LOW_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*LOW_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q +
     a7*b7*LOW_W*HIGH_Z*LOW_V*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*MED_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*MED_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*MED_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q +
     a7*b7*MED_W*HIGH_Z*LOW_V*HIGH_Q;
    IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*LOW_V + a4*b4*HIGH_W*LOW_V + a5*b4*HIGH_Z*LOW_V +
     a7*b4*HIGH_W*HIGH_Z*LOW_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q +
a1*b7*LOW_V*HIGH_Q +
     a4*b7*HIGH_W*LOW_V*HIGH_Q + a5*b7*HIGH_Z*LOW_V*HIGH_Q +
     a7*b7*HIGH_W*HIGH_Z*LOW_V*HIGH_Q;

    ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*LOW_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q +
     a7*b7*LOW_W*LOW_Z*MED_V*HIGH_Q;
    IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*MED_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
```

```
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q +
     a7*b7*MED_W*LOW_Z*MED_V*HIGH_Q;
    IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*LOW_Z*MED_V +
     a7*b4*HIGH_W*LOW_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*LOW_Z*MED_V*HIGH_Q +
     a7*b7*HIGH_W*LOW_Z*MED_V*HIGH_Q;

    ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*LOW_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q +
     a7*b7*LOW_W*MED_Z*MED_V*HIGH_Q;
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*MED_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q +
     a7*b7*MED_W*MED_Z*MED_V*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*MED_Z*MED_V +
     a7*b4*HIGH_W*MED_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*MED_Z*MED_V*HIGH_Q +
     a7*b7*HIGH_W*MED_Z*MED_V*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*LOW_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*LOW_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*LOW_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q +
```

```
     a7*b7*LOW_W*HIGH_Z*MED_V*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*MED_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*MED_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*MED_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q +
     a7*b7*MED_W*HIGH_Z*MED_V*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*MED_V + a4*b4*HIGH_W*MED_V + a5*b4*HIGH_Z*MED_V +
     a7*b4*HIGH_W*HIGH_Z*MED_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q +
a1*b7*MED_V*HIGH_Q +
     a4*b7*HIGH_W*MED_V*HIGH_Q + a5*b7*HIGH_Z*MED_V*HIGH_Q +
     a7*b7*HIGH_W*HIGH_Z*MED_V*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*LOW_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*LOW_W*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q +
     a7*b7*LOW_W*LOW_Z*HIGH_V*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*LOW_Z*HIGH_V +
     a7*b4*MED_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*MED_W*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q +
     a7*b7*MED_W*LOW_Z*HIGH_V*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*LOW_Z*HIGH_V
+
     a7*b4*HIGH_W*LOW_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*LOW_Z*HIGH_Q + a7*b5*HIGH_W*LOW_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*LOW_Z*HIGH_V*HIGH_Q +
     a7*b7*HIGH_W*LOW_Z*HIGH_V*HIGH_Q;
```

```
    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*LOW_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*LOW_W*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q +
     a7*b7*LOW_W*MED_Z*HIGH_V*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*MED_Z*HIGH_V +
     a7*b4*MED_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*MED_W*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q +
     a7*b7*MED_W*MED_Z*HIGH_V*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*MED_Z*HIGH_V
+
     a7*b4*HIGH_W*MED_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*MED_Z*HIGH_Q + a7*b5*HIGH_W*MED_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*MED_Z*HIGH_V*HIGH_Q +
     a7*b7*HIGH_W*MED_Z*HIGH_V*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*LOW_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*LOW_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*LOW_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*LOW_W*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*LOW_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q +
     a7*b7*LOW_W*HIGH_Z*HIGH_V*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*MED_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*MED_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*MED_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*MED_W*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*MED_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q +
```

```
     a7*b7*MED_W*HIGH_Z*HIGH_V*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_V + a4*b4*HIGH_W*HIGH_V + a5*b4*HIGH_Z*HIGH_V
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V + a1*b5*HIGH_Q +
a4*b5*HIGH_W*HIGH_Q +
     a5*b5*HIGH_Z*HIGH_Q + a7*b5*HIGH_W*HIGH_Z*HIGH_Q +
a1*b7*HIGH_V*HIGH_Q +
     a4*b7*HIGH_W*HIGH_V*HIGH_Q + a5*b7*HIGH_Z*HIGH_V*HIGH_Q
+
     a7*b7*HIGH_W*HIGH_Z*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + DLOW_LOZ;
    TMLLL = IMLLL + DMEW_LOZ;
    THLLL = IHLLL + DHIW_LOZ;

    TLMLL = ILMLL + DLOW_MEZ;
    TMMLL = IMMLL + DMEW_MEZ;
    THMLL = IHMLL + DHIW_MEZ;

    TLHLL = ILHLL + DLOW_HIZ;
    TMHLL = IMHLL + DMEW_HIZ;
    THHLL = IHHLL + DHIW_HIZ;
```

```
TLLML = ILLML + DLOW_LOZ;
TMLML = IMLML + DMEW_LOZ;
THLML = IHLML + DHIW_LOZ;

TLMML = ILMML + DLOW_MEZ;
TMMML = IMMML + DMEW_MEZ;
THMML = IHMML + DHIW_MEZ;

TLHML = ILHML + DLOW_HIZ;
TMHML = IMHML + DMEW_HIZ;
THHML = IHHML + DHIW_HIZ;

TLLHL = ILLHL + DLOW_LOZ;
TMLHL = IMLHL + DMEW_LOZ;
THLHL = IHLHL + DHIW_LOZ;

TLMHL = ILMHL + DLOW_MEZ;
TMMHL = IMMHL + DMEW_MEZ;
THMHL = IHMHL + DHIW_MEZ;

TLHHL = ILHHL + DLOW_HIZ;
TMHHL = IMHHL + DMEW_HIZ;
THHHL = IHHHL + DHIW_HIZ;

TLLLM = ILLLM + DLOW_LOZ;
TMLLM = IMLLM + DMEW_LOZ;
THLLM = IHLLM + DHIW_LOZ;

TLMLM = ILMLM + DLOW_MEZ;
TMMLM = IMMLM + DMEW_MEZ;
THMLM = IHMLM + DHIW_MEZ;

TLHLM = ILHLM + DLOW_HIZ;
TMHLM = IMHLM + DMEW_HIZ;
THHLM = IHHLM + DHIW_HIZ;

TLLMM = ILLMM + DLOW_LOZ;
TMLMM = IMLMM + DMEW_LOZ;
THLMM = IHLMM + DHIW_LOZ;

TLMMM = ILMMM + DLOW_MEZ;
TMMMM = IMMMM + DMEW_MEZ;
THMMM = IHMMM + DHIW_MEZ;

TLHMM = ILHMM + DLOW_HIZ;
TMHMM = IMHMM + DMEW_HIZ;
THHMM = IHHMM + DHIW_HIZ;

TLLHM = ILLHM + DLOW_LOZ;
TMLHM = IMLHM + DMEW_LOZ;
THLHM = IHLHM + DHIW_LOZ;
```

```
    TLMHM = ILMHM + DLOW_MEZ;
    TMMHM = IMMHM + DMEW_MEZ;
    THMHM = IHMHM + DHIW_MEZ;

    TLHHM = ILHHM + DLOW_HIZ;
    TMHHM = IMHHM + DMEW_HIZ;
    THHHM = IHHHM + DHIW_HIZ;

    TLLLH = ILLLH + DLOW_LOZ;
    TMLLH = IMLLH + DMEW_LOZ;
    THLLH = IHLLH + DHIW_LOZ;

    TLMLH = ILMLH + DLOW_MEZ;
    TMMLH = IMMLH + DMEW_MEZ;
    THMLH = IHMLH + DHIW_MEZ;

    TLHLH = ILHLH + DLOW_HIZ;
    TMHLH = IMHLH + DMEW_HIZ;
    THHLH = IHHLH + DHIW_HIZ;

    TLLMH = ILLMH + DLOW_LOZ;
    TMLMH = IMLMH + DMEW_LOZ;
    THLMH = IHLMH + DHIW_LOZ;

    TLMMH = ILMMH + DLOW_MEZ;
    TMMMH = IMMMH + DMEW_MEZ;
    THMMH = IHMMH + DHIW_MEZ;

    TLHMH = ILHMH + DLOW_HIZ;
    TMHMH = IMHMH + DMEW_HIZ;
    THHMH = IHHMH + DHIW_HIZ;

    TLLHH = ILLHH + DLOW_LOZ;
    TMLHH = IMLHH + DMEW_LOZ;
    THLHH = IHLHH + DHIW_LOZ;

    TLMHH = ILMHH + DLOW_MEZ;
    TMMHH = IMMHH + DMEW_MEZ;
    THMHH = IHMHH + DHIW_MEZ;

    TLHHH = ILHHH + DLOW_HIZ;
    TMHHH = IMHHH + DMEW_HIZ;
    THHHH = IHHHH + DHIW_HIZ;
! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis
```

```
    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;

    PLMLL = ILMLL*XVAL;
    PMMLL = IMMLL*XVAL;
    PHMLL = IHMLL*XVAL;

    PLHLL = ILHLL*XVAL;
    PMHLL = IMHLL*XVAL;
    PHHLL = IHHLL*XVAL;

    PLLML = ILLML*XVAL;
    PMLML = IMLML*XVAL;
    PHLML = IHLML*XVAL;

    PLMML = ILMML*XVAL;
    PMMML = IMMML*XVAL;
    PHMML = IHMML*XVAL;

    PLHML = ILHML*XVAL;
    PMHML = IMHML*XVAL;
    PHHML = IHHML*XVAL;

    PLLHL = ILLHL*XVAL;
    PMLHL = IMLHL*XVAL;
    PHLHL = IHLHL*XVAL;

    PLMHL = ILMHL*XVAL;
    PMMHL = IMMHL*XVAL;
    PHMHL = IHMHL*XVAL;

    PLHHL = ILHHL*XVAL;
    PMHHL = IMHHL*XVAL;
    PHHHL = IHHHL*XVAL;

    PLLLM = ILLLM*XVAL;
    PMLLM = IMLLM*XVAL;
    PHLLM = IHLLM*XVAL;
```

```
PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;

PLHLH = ILHLH*XVAL;
PMHLH = IMHLH*XVAL;
PHHLH = IHHLH*XVAL;

PLLMH = ILLMH*XVAL;
PMLMH = IMLMH*XVAL;
PHLMH = IHLMH*XVAL;

PLMMH = ILMMH*XVAL;
PMMMH = IMMMH*XVAL;
PHMMH = IHMMH*XVAL;
```

```
    PLHMH = ILHMH*XVAL;
    PMHMH = IMHMH*XVAL;
    PHHMH = IHHMH*XVAL;

    PLLHH = ILLHH*XVAL;
    PMLHH = IMLHH*XVAL;
    PHLHH = IHLHH*XVAL;

    PLMHH = ILMHH*XVAL;
    PMMHH = IMMHH*XVAL;
    PHMHH = IHMHH*XVAL;

    PLHHH = ILHHH*XVAL;
    PMHHH = IMHHH*XVAL;
    PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 56: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate the IV-Mediator path with all 2-way and 3-way interactions, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path with all 2-way and 3-way interactions

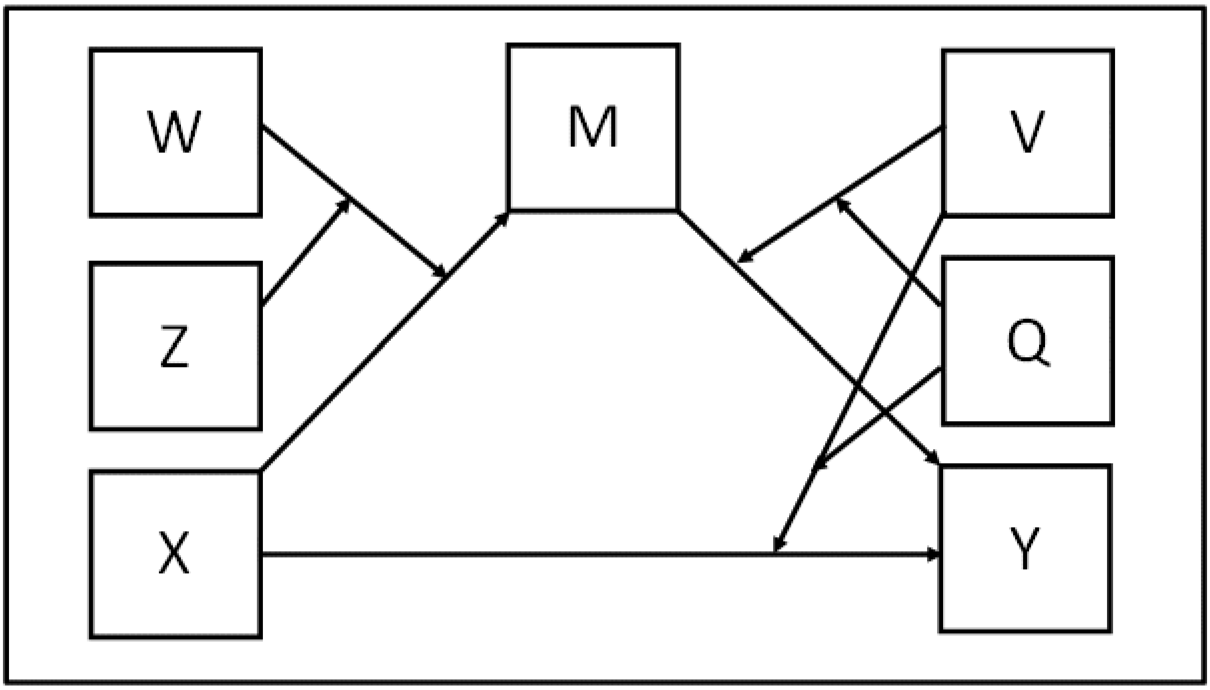
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

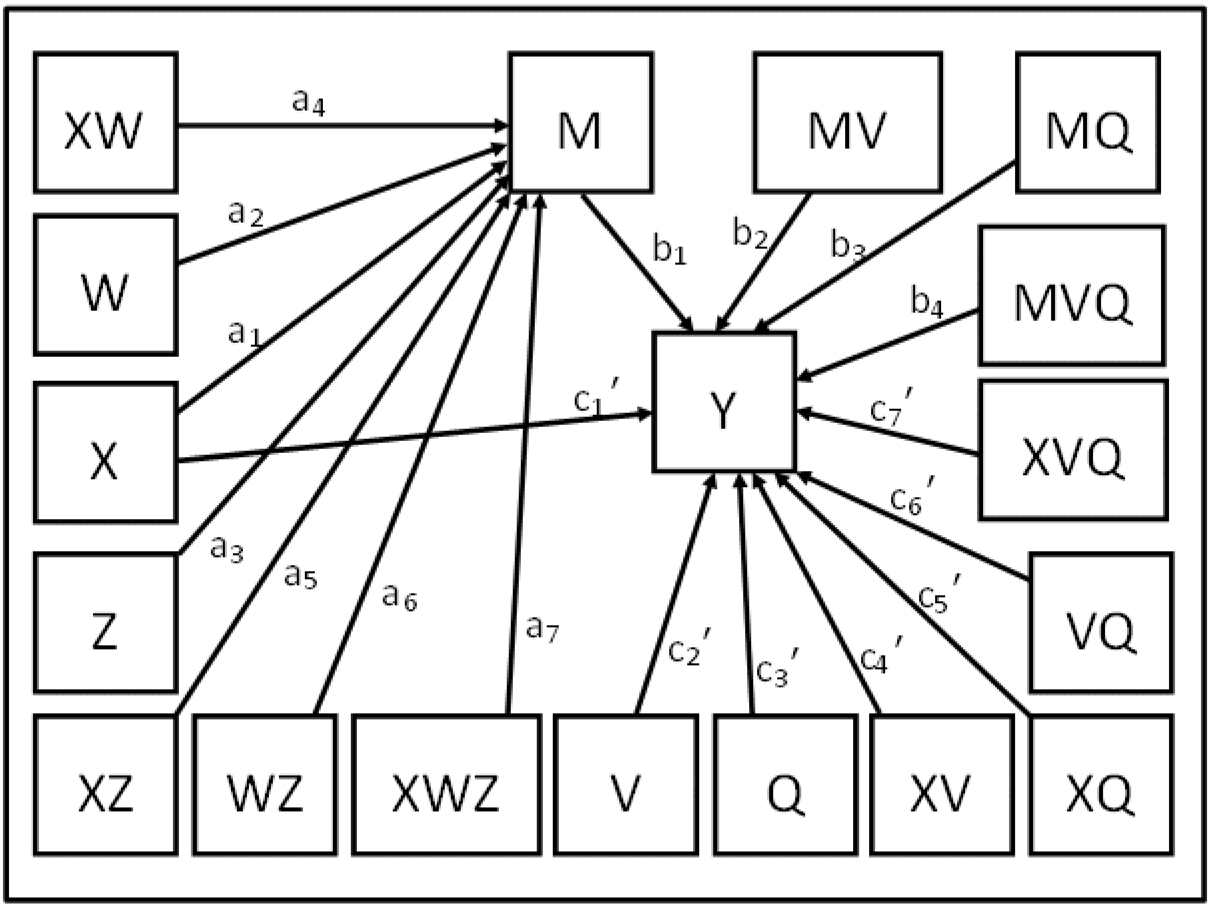
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + b3(a0 + a1X + a2W + a3Z

+ a4XW + a5XZ + a6WZ + a7XWZ)Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)VQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + a0b2V + a1b2XV + a2b2WV + a3b2ZV + a4b2XWV + a5b2XZV + a6b2WZV + a7b2XWZV + a0b3Q + a1b3XQ + a2b3WQ + a3b3ZQ + a4b3XWQ + a5b3XZQ + a6b3WZQ + a7b3XWZQ + a0b4VQ + a1b4XVQ + a2b4WVQ + a3b4ZVQ + a4b4XWVQ + a5b4XZVQ + a6b4WZVQ + a7b4XWZVQ + c1'X + c2'V + c3'Q + c4'XV + c5'XQ + c6'VQ + c7'XVQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + a0b2V + a2b2WV + a3b2ZV + a6b2WZV + a0b3Q + a2b3WQ + a3b3ZQ + a6b3WZQ + a0b4VQ + a2b4WVQ + a3b4ZVQ + a6b4WZVQ + c2'V + c3'Q + c6'VQ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV + a1b3Q + a4b3WQ + a5b3ZQ + a7b3WZQ + a1b4VQ + a4b4WVQ + a5b4ZVQ + a7b4WZVQ + c1' + c4'V + c5'Q + c7'VQ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV + a1b3Q + a4b3WQ + a5b3ZQ + a7b3WZQ + a1b4VQ + a4b4WVQ + a5b4ZVQ + a7b4WZVQ = (a1 + a4W + a5Z + a7WZ)(b1 + b2V + b3Q + b4VQ)

One direct effect of X on Y, conditional on V, Q:

c1' + c4'V + c5'Q + c7'VQ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ WZ XV XQ VQ MV MQ XWZ XVQ
MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MQ = M*Q;
```

```
    MV = M*V;
    XW = X*W;
    XZ = X*Z;
    XQ = X*Q;
    XV = X*V;
    WZ = W*Z;
    VQ = V*Q;
    MVQ = M*V*Q;
    XWZ = X*W*Z;
    XVQ = X*V*Q;

ANALYSIS:
    TYPE = GENERAL;
    ESTIMATOR = ML;
    BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON MV (b2);
    Y ON MQ (b3);
    Y ON MVQ (b4);

    Y ON X (cdash1);
    Y ON V (cdash2);
    Y ON Q (cdash3);
    Y ON XV (cdash4);
    Y ON XQ (cdash5);
    Y ON VQ (cdash6);
    Y ON XVQ (cdash7);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
    M ON WZ (a6);
    M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean
```

```
! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    DLOV_LOQ DMEV_LOQ DHIV_LOQ DLOV_MEQ DMEV_MEQ DHIV_MEQ
    DLOV_HIQ DMEV_HIQ DHIV_HIQ
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
```

```
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*LOW_W*LOW_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*LOW_W*LOW_V*LOW_Q + a5*b4*LOW_Z*LOW_V*LOW_Q +
     a7*b4*LOW_W*LOW_Z*LOW_V*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*MED_W*LOW_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*MED_W*LOW_V*LOW_Q + a5*b4*LOW_Z*LOW_V*LOW_Q +
     a7*b4*MED_W*LOW_Z*LOW_V*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*HIGH_W*LOW_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*HIGH_W*LOW_V*LOW_Q + a5*b4*LOW_Z*LOW_V*LOW_Q +
     a7*b4*HIGH_W*LOW_Z*LOW_V*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*LOW_W*MED_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*LOW_W*LOW_V*LOW_Q + a5*b4*MED_Z*LOW_V*LOW_Q +
```

```
     a7*b4*LOW_W*MED_Z*LOW_V*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*MED_W*MED_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*MED_W*LOW_V*LOW_Q + a5*b4*MED_Z*LOW_V*LOW_Q +
     a7*b4*MED_W*MED_Z*LOW_V*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*HIGH_W*MED_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*HIGH_W*LOW_V*LOW_Q + a5*b4*MED_Z*LOW_V*LOW_Q +
     a7*b4*HIGH_W*MED_Z*LOW_V*LOW_Q;

    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*LOW_W*HIGH_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*LOW_W*LOW_V*LOW_Q + a5*b4*HIGH_Z*LOW_V*LOW_Q +
     a7*b4*LOW_W*HIGH_Z*LOW_V*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*MED_W*HIGH_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*MED_W*LOW_V*LOW_Q + a5*b4*HIGH_Z*LOW_V*LOW_Q +
     a7*b4*MED_W*HIGH_Z*LOW_V*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*HIGH_W*HIGH_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*HIGH_W*LOW_V*LOW_Q + a5*b4*HIGH_Z*LOW_V*LOW_Q +
     a7*b4*HIGH_W*HIGH_Z*LOW_V*LOW_Q;
```

```
    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*LOW_W*LOW_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*LOW_W*MED_V*LOW_Q + a5*b4*LOW_Z*MED_V*LOW_Q +
     a7*b4*LOW_W*LOW_Z*MED_V*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*MED_W*LOW_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*MED_W*MED_V*LOW_Q + a5*b4*LOW_Z*MED_V*LOW_Q +
     a7*b4*MED_W*LOW_Z*MED_V*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*HIGH_W*LOW_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*HIGH_W*MED_V*LOW_Q + a5*b4*LOW_Z*MED_V*LOW_Q +
     a7*b4*HIGH_W*LOW_Z*MED_V*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*LOW_W*MED_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*LOW_W*MED_V*LOW_Q + a5*b4*MED_Z*MED_V*LOW_Q +
     a7*b4*LOW_W*MED_Z*MED_V*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*MED_W*MED_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*MED_W*MED_V*LOW_Q + a5*b4*MED_Z*MED_V*LOW_Q +
     a7*b4*MED_W*MED_Z*MED_V*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
```

```
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*HIGH_W*MED_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*HIGH_W*MED_V*LOW_Q + a5*b4*MED_Z*MED_V*LOW_Q +
     a7*b4*HIGH_W*MED_Z*MED_V*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*LOW_W*HIGH_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*LOW_W*MED_V*LOW_Q + a5*b4*HIGH_Z*MED_V*LOW_Q +
     a7*b4*LOW_W*HIGH_Z*MED_V*LOW_Q;
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*MED_W*HIGH_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*MED_W*MED_V*LOW_Q + a5*b4*HIGH_Z*MED_V*LOW_Q +
     a7*b4*MED_W*HIGH_Z*MED_V*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*HIGH_W*HIGH_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*HIGH_W*MED_V*LOW_Q + a5*b4*HIGH_Z*MED_V*LOW_Q +
     a7*b4*HIGH_W*HIGH_Z*MED_V*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*LOW_W*LOW_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*LOW_W*HIGH_V*LOW_Q + a5*b4*LOW_Z*HIGH_V*LOW_Q +
     a7*b4*LOW_W*LOW_Z*HIGH_V*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
```

```
     a7*b2*MED_W*LOW_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*MED_W*LOW_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*MED_W*HIGH_V*LOW_Q + a5*b4*LOW_Z*HIGH_V*LOW_Q +
     a7*b4*MED_W*LOW_Z*HIGH_V*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*HIGH_W*LOW_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*HIGH_W*HIGH_V*LOW_Q + a5*b4*LOW_Z*HIGH_V*LOW_Q +
     a7*b4*HIGH_W*LOW_Z*HIGH_V*LOW_Q;

    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*LOW_W*MED_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*LOW_W*HIGH_V*LOW_Q + a5*b4*MED_Z*HIGH_V*LOW_Q +
     a7*b4*LOW_W*MED_Z*HIGH_V*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*MED_W*MED_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*MED_W*HIGH_V*LOW_Q + a5*b4*MED_Z*HIGH_V*LOW_Q +
     a7*b4*MED_W*MED_Z*HIGH_V*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*HIGH_W*MED_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*HIGH_W*HIGH_V*LOW_Q + a5*b4*MED_Z*HIGH_V*LOW_Q +
     a7*b4*HIGH_W*MED_Z*HIGH_V*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
```

```
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*LOW_W*HIGH_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*LOW_W*HIGH_V*LOW_Q + a5*b4*HIGH_Z*HIGH_V*LOW_Q +
     a7*b4*LOW_W*HIGH_Z*HIGH_V*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*MED_W*HIGH_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*MED_W*HIGH_V*LOW_Q + a5*b4*HIGH_Z*HIGH_V*LOW_Q +
     a7*b4*MED_W*HIGH_Z*HIGH_V*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*HIGH_W*HIGH_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*HIGH_W*HIGH_V*LOW_Q + a5*b4*HIGH_Z*HIGH_V*LOW_Q +
     a7*b4*HIGH_W*HIGH_Z*HIGH_V*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*LOW_W*LOW_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*LOW_W*LOW_V*MED_Q + a5*b4*LOW_Z*LOW_V*MED_Q +
     a7*b4*LOW_W*LOW_Z*LOW_V*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*MED_W*LOW_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*MED_W*LOW_V*MED_Q + a5*b4*LOW_Z*LOW_V*MED_Q +
     a7*b4*MED_W*LOW_Z*LOW_V*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
```

```
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*HIGH_W*LOW_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*HIGH_W*LOW_V*MED_Q + a5*b4*LOW_Z*LOW_V*MED_Q +
     a7*b4*HIGH_W*LOW_Z*LOW_V*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*LOW_W*MED_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*LOW_W*LOW_V*MED_Q + a5*b4*MED_Z*LOW_V*MED_Q +
     a7*b4*LOW_W*MED_Z*LOW_V*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*MED_W*MED_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*MED_W*LOW_V*MED_Q + a5*b4*MED_Z*LOW_V*MED_Q +
     a7*b4*MED_W*MED_Z*LOW_V*MED_Q;
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*HIGH_W*MED_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*HIGH_W*LOW_V*MED_Q + a5*b4*MED_Z*LOW_V*MED_Q +
     a7*b4*HIGH_W*MED_Z*LOW_V*MED_Q;

    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*LOW_W*HIGH_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*LOW_W*LOW_V*MED_Q + a5*b4*HIGH_Z*LOW_V*MED_Q +
     a7*b4*LOW_W*HIGH_Z*LOW_V*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
```

```
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*MED_W*HIGH_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*MED_W*LOW_V*MED_Q + a5*b4*HIGH_Z*LOW_V*MED_Q +
     a7*b4*MED_W*HIGH_Z*LOW_V*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*HIGH_W*HIGH_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*HIGH_W*LOW_V*MED_Q + a5*b4*HIGH_Z*LOW_V*MED_Q +
     a7*b4*HIGH_W*HIGH_Z*LOW_V*MED_Q;

    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*LOW_W*LOW_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*LOW_W*MED_V*MED_Q + a5*b4*LOW_Z*MED_V*MED_Q +
     a7*b4*LOW_W*LOW_Z*MED_V*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*MED_W*LOW_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*MED_W*MED_V*MED_Q + a5*b4*LOW_Z*MED_V*MED_Q +
     a7*b4*MED_W*LOW_Z*MED_V*MED_Q;
    IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*HIGH_W*LOW_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*HIGH_W*MED_V*MED_Q + a5*b4*LOW_Z*MED_V*MED_Q +
     a7*b4*HIGH_W*LOW_Z*MED_V*MED_Q;

    ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
```

```
     a7*b2*LOW_W*MED_Z*MED_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*LOW_W*MED_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*LOW_W*MED_V*MED_Q + a5*b4*MED_Z*MED_V*MED_Q +
     a7*b4*LOW_W*MED_Z*MED_V*MED_Q;
    IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*MED_W*MED_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*MED_W*MED_V*MED_Q + a5*b4*MED_Z*MED_V*MED_Q +
     a7*b4*MED_W*MED_Z*MED_V*MED_Q;
    IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*HIGH_W*MED_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*HIGH_W*MED_V*MED_Q + a5*b4*MED_Z*MED_V*MED_Q +
     a7*b4*HIGH_W*MED_Z*MED_V*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*LOW_W*HIGH_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*LOW_W*MED_V*MED_Q + a5*b4*HIGH_Z*MED_V*MED_Q +
     a7*b4*LOW_W*HIGH_Z*MED_V*MED_Q;
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*MED_W*HIGH_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*MED_W*MED_V*MED_Q + a5*b4*HIGH_Z*MED_V*MED_Q +
     a7*b4*MED_W*HIGH_Z*MED_V*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
```

```
     a5*b3*HIGH_Z*MED_Q + a7*b3*HIGH_W*HIGH_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*HIGH_W*MED_V*MED_Q + a5*b4*HIGH_Z*MED_V*MED_Q +
     a7*b4*HIGH_W*HIGH_Z*MED_V*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*LOW_W*LOW_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*LOW_W*HIGH_V*MED_Q + a5*b4*LOW_Z*HIGH_V*MED_Q +
     a7*b4*LOW_W*LOW_Z*HIGH_V*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*MED_W*LOW_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*MED_W*HIGH_V*MED_Q + a5*b4*LOW_Z*HIGH_V*MED_Q +
     a7*b4*MED_W*LOW_Z*HIGH_V*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*HIGH_W*LOW_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*HIGH_W*HIGH_V*MED_Q + a5*b4*LOW_Z*HIGH_V*MED_Q +
     a7*b4*HIGH_W*LOW_Z*HIGH_V*MED_Q;

    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*LOW_W*MED_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*LOW_W*HIGH_V*MED_Q + a5*b4*MED_Z*HIGH_V*MED_Q +
     a7*b4*LOW_W*MED_Z*HIGH_V*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
```

```
     a5*b3*MED_Z*MED_Q + a7*b3*MED_W*MED_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*MED_W*HIGH_V*MED_Q + a5*b4*MED_Z*HIGH_V*MED_Q +
     a7*b4*MED_W*MED_Z*HIGH_V*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*HIGH_W*MED_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*HIGH_W*HIGH_V*MED_Q + a5*b4*MED_Z*HIGH_V*MED_Q +
     a7*b4*HIGH_W*MED_Z*HIGH_V*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*LOW_W*HIGH_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*LOW_W*HIGH_V*MED_Q + a5*b4*HIGH_Z*HIGH_V*MED_Q +
     a7*b4*LOW_W*HIGH_Z*HIGH_V*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*MED_W*HIGH_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*MED_W*HIGH_V*MED_Q + a5*b4*HIGH_Z*HIGH_V*MED_Q +
     a7*b4*MED_W*HIGH_Z*HIGH_V*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*HIGH_W*HIGH_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*HIGH_W*HIGH_V*MED_Q + a5*b4*HIGH_Z*HIGH_V*MED_Q +
     a7*b4*HIGH_W*HIGH_Z*HIGH_V*MED_Q;

    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
```

```
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*LOW_W*LOW_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*LOW_W*LOW_V*HIGH_Q + a5*b4*LOW_Z*LOW_V*HIGH_Q +
     a7*b4*LOW_W*LOW_Z*LOW_V*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*MED_W*LOW_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*MED_W*LOW_V*HIGH_Q + a5*b4*LOW_Z*LOW_V*HIGH_Q +
     a7*b4*MED_W*LOW_Z*LOW_V*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*HIGH_W*LOW_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*HIGH_W*LOW_V*HIGH_Q + a5*b4*LOW_Z*LOW_V*HIGH_Q +
     a7*b4*HIGH_W*LOW_Z*LOW_V*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*LOW_W*MED_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*LOW_W*LOW_V*HIGH_Q + a5*b4*MED_Z*LOW_V*HIGH_Q +
     a7*b4*LOW_W*MED_Z*LOW_V*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*MED_W*MED_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*MED_W*LOW_V*HIGH_Q + a5*b4*MED_Z*LOW_V*HIGH_Q +
     a7*b4*MED_W*MED_Z*LOW_V*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V + a1*b3*HIGH_Q +
```

```
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*HIGH_W*MED_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*HIGH_W*LOW_V*HIGH_Q + a5*b4*MED_Z*LOW_V*HIGH_Q +
     a7*b4*HIGH_W*MED_Z*LOW_V*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*LOW_W*HIGH_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*LOW_W*LOW_V*HIGH_Q + a5*b4*HIGH_Z*LOW_V*HIGH_Q +
     a7*b4*LOW_W*HIGH_Z*LOW_V*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*MED_W*HIGH_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*MED_W*LOW_V*HIGH_Q + a5*b4*HIGH_Z*LOW_V*HIGH_Q +
     a7*b4*MED_W*HIGH_Z*LOW_V*HIGH_Q;
    IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*HIGH_W*HIGH_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*HIGH_W*LOW_V*HIGH_Q + a5*b4*HIGH_Z*LOW_V*HIGH_Q +
     a7*b4*HIGH_W*HIGH_Z*LOW_V*HIGH_Q;

    ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*LOW_W*LOW_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*LOW_W*MED_V*HIGH_Q + a5*b4*LOW_Z*MED_V*HIGH_Q +
     a7*b4*LOW_W*LOW_Z*MED_V*HIGH_Q;
    IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
```

```
     a5*b3*LOW_Z*HIGH_Q + a7*b3*MED_W*LOW_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*MED_W*MED_V*HIGH_Q + a5*b4*LOW_Z*MED_V*HIGH_Q +
     a7*b4*MED_W*LOW_Z*MED_V*HIGH_Q;
    IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*HIGH_W*LOW_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*HIGH_W*MED_V*HIGH_Q + a5*b4*LOW_Z*MED_V*HIGH_Q +
     a7*b4*HIGH_W*LOW_Z*MED_V*HIGH_Q;

    ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*LOW_W*MED_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*LOW_W*MED_V*HIGH_Q + a5*b4*MED_Z*MED_V*HIGH_Q +
     a7*b4*LOW_W*MED_Z*MED_V*HIGH_Q;
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*MED_W*MED_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*MED_W*MED_V*HIGH_Q + a5*b4*MED_Z*MED_V*HIGH_Q +
     a7*b4*MED_W*MED_Z*MED_V*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*HIGH_W*MED_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*HIGH_W*MED_V*HIGH_Q + a5*b4*MED_Z*MED_V*HIGH_Q +
     a7*b4*HIGH_W*MED_Z*MED_V*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*LOW_W*HIGH_Z*HIGH_Q +
```

```
a1*b4*MED_V*HIGH_Q +
     a4*b4*LOW_W*MED_V*HIGH_Q + a5*b4*HIGH_Z*MED_V*HIGH_Q +
     a7*b4*LOW_W*HIGH_Z*MED_V*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*MED_W*HIGH_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*MED_W*MED_V*HIGH_Q + a5*b4*HIGH_Z*MED_V*HIGH_Q +
     a7*b4*MED_W*HIGH_Z*MED_V*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*HIGH_W*HIGH_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*HIGH_W*MED_V*HIGH_Q + a5*b4*HIGH_Z*MED_V*HIGH_Q +
     a7*b4*HIGH_W*HIGH_Z*MED_V*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*LOW_W*LOW_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*LOW_W*HIGH_V*HIGH_Q + a5*b4*LOW_Z*HIGH_V*HIGH_Q +
     a7*b4*LOW_W*LOW_Z*HIGH_V*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*MED_W*LOW_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*MED_W*HIGH_V*HIGH_Q + a5*b4*LOW_Z*HIGH_V*HIGH_Q +
     a7*b4*MED_W*LOW_Z*HIGH_V*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*HIGH_W*LOW_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
```

```
     a4*b4*HIGH_W*HIGH_V*HIGH_Q + a5*b4*LOW_Z*HIGH_V*HIGH_Q +
     a7*b4*HIGH_W*LOW_Z*HIGH_V*HIGH_Q;

    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*LOW_W*MED_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*LOW_W*HIGH_V*HIGH_Q + a5*b4*MED_Z*HIGH_V*HIGH_Q +
     a7*b4*LOW_W*MED_Z*HIGH_V*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*MED_W*MED_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*MED_W*HIGH_V*HIGH_Q + a5*b4*MED_Z*HIGH_V*HIGH_Q +
     a7*b4*MED_W*MED_Z*HIGH_V*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*HIGH_W*MED_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*HIGH_W*HIGH_V*HIGH_Q + a5*b4*MED_Z*HIGH_V*HIGH_Q +
     a7*b4*HIGH_W*MED_Z*HIGH_V*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*LOW_W*HIGH_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*LOW_W*HIGH_V*HIGH_Q + a5*b4*HIGH_Z*HIGH_V*HIGH_Q +
     a7*b4*LOW_W*HIGH_Z*HIGH_V*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
```

```
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*MED_W*HIGH_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*MED_W*HIGH_V*HIGH_Q + a5*b4*HIGH_Z*HIGH_V*HIGH_Q +
     a7*b4*MED_W*HIGH_Z*HIGH_V*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*HIGH_W*HIGH_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*HIGH_W*HIGH_V*HIGH_Q + a5*b4*HIGH_Z*HIGH_V*HIGH_Q
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLOV_LOQ = cdash1 + cdash4*LOW_V + cdash5*LOW_Q +
cdash7*LOW_V*LOW_Q;
    DMEV_LOQ = cdash1 + cdash4*MED_W + cdash5*LOW_Q +
cdash7*MED_W*LOW_Q;
    DHIV_LOQ = cdash1 + cdash4*HIGH_V + cdash5*LOW_Q +
cdash7*HIGH_V*LOW_Q;

    DLOV_MEQ = cdash1 + cdash4*LOW_V + cdash5*MED_Q +
cdash7*LOW_V*MED_Q;
    DMEV_MEQ = cdash1 + cdash4*MED_W + cdash5*MED_Q +
cdash7*MED_W*MED_Q;
    DHIV_MEQ = cdash1 + cdash4*HIGH_V + cdash5*MED_Q +
cdash7*HIGH_V*MED_Q;

    DLOV_HIQ = cdash1 + cdash4*LOW_V + cdash5*HIGH_Q +
cdash7*LOW_V*HIGH_Q;
    DMEV_HIQ = cdash1 + cdash4*MED_W + cdash5*HIGH_Q +
cdash7*MED_W*HIGH_Q;
    DHIV_HIQ = cdash1 + cdash4*HIGH_V + cdash5*HIGH_Q +
cdash7*HIGH_V*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + DLOV_LOQ;
    TMLLL = IMLLL + DLOV_LOQ;
    THLLL = IHLLL + DLOV_LOQ;

    TLMLL = ILMLL + DLOV_LOQ;
    TMMLL = IMMLL + DLOV_LOQ;
    THMLL = IHMLL + DLOV_LOQ;
```

```
TLHLL = ILHLL + DLOV_LOQ;
TMHLL = IMHLL + DLOV_LOQ;
THHLL = IHHLL + DLOV_LOQ;

TLLML = ILLML + DMEV_LOQ;
TMLML = IMLML + DMEV_LOQ;
THLML = IHLML + DMEV_LOQ;

TLMML = ILMML + DMEV_LOQ;
TMMML = IMMML + DMEV_LOQ;
THMML = IHMML + DMEV_LOQ;

TLHML = ILHML + DMEV_LOQ;
TMHML = IMHML + DMEV_LOQ;
THHML = IHHML + DMEV_LOQ;

TLLHL = ILLHL + DHIV_LOQ;
TMLHL = IMLHL + DHIV_LOQ;
THLHL = IHLHL + DHIV_LOQ;

TLMHL = ILMHL + DHIV_LOQ;
TMMHL = IMMHL + DHIV_LOQ;
THMHL = IHMHL + DHIV_LOQ;

TLHHL = ILHHL + DHIV_LOQ;
TMHHL = IMHHL + DHIV_LOQ;
THHHL = IHHHL + DHIV_LOQ;

TLLLM = ILLLM + DLOV_MEQ;
TMLLM = IMLLM + DLOV_MEQ;
THLLM = IHLLM + DLOV_MEQ;

TLMLM = ILMLM + DLOV_MEQ;
TMMLM = IMMLM + DLOV_MEQ;
THMLM = IHMLM + DLOV_MEQ;

TLHLM = ILHLM + DLOV_MEQ;
TMHLM = IMHLM + DLOV_MEQ;
THHLM = IHHLM + DLOV_MEQ;

TLLMM = ILLMM + DMEV_MEQ;
TMLMM = IMLMM + DMEV_MEQ;
THLMM = IHLMM + DMEV_MEQ;

TLMMM = ILMMM + DMEV_MEQ;
TMMMM = IMMMM + DMEV_MEQ;
THMMM = IHMMM + DMEV_MEQ;

TLHMM = ILHMM + DMEV_MEQ;
TMHMM = IMHMM + DMEV_MEQ;
THHMM = IHHMM + DMEV_MEQ;
```

```
    TLLHM = ILLHM + DHIV_MEQ;
    TMLHM = IMLHM + DHIV_MEQ;
    THLHM = IHLHM + DHIV_MEQ;

    TLMHM = ILMHM + DHIV_MEQ;
    TMMHM = IMMHM + DHIV_MEQ;
    THMHM = IHMHM + DHIV_MEQ;

    TLHHM = ILHHM + DHIV_MEQ;
    TMHHM = IMHHM + DHIV_MEQ;
    THHHM = IHHHM + DHIV_MEQ;

    TLLLH = ILLLH + DLOV_HIQ;
    TMLLH = IMLLH + DLOV_HIQ;
    THLLH = IHLLH + DLOV_HIQ;

    TLMLH = ILMLH + DLOV_HIQ;
    TMMLH = IMMLH + DLOV_HIQ;
    THMLH = IHMLH + DLOV_HIQ;

    TLHLH = ILHLH + DLOV_HIQ;
    TMHLH = IMHLH + DLOV_HIQ;
    THHLH = IHHLH + DLOV_HIQ;

    TLLMH = ILLMH + DMEV_HIQ;
    TMLMH = IMLMH + DMEV_HIQ;
    THLMH = IHLMH + DMEV_HIQ;

    TLMMH = ILMMH + DMEV_HIQ;
    TMMMH = IMMMH + DMEV_HIQ;
    THMMH = IHMMH + DMEV_HIQ;

    TLHMH = ILHMH + DMEV_HIQ;
    TMHMH = IMHMH + DMEV_HIQ;
    THHMH = IHHMH + DMEV_HIQ;

    TLLHH = ILLHH + DHIV_HIQ;
    TMLHH = IMLHH + DHIV_HIQ;
    THLHH = IHLHH + DHIV_HIQ;

    TLMHH = ILMHH + DHIV_HIQ;
    TMMHH = IMMHH + DHIV_HIQ;
    THMHH = IHMHH + DHIV_HIQ;

    TLHHH = ILHHH + DHIV_HIQ;
    TMHHH = IMHHH + DHIV_HIQ;
    THHHH = IHHHH + DHIV_HIQ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
```

```
by
! logical min and max limits of predictor X used in analysis
    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;

    PLMLL = ILMLL*XVAL;
    PMMLL = IMMLL*XVAL;
    PHMLL = IHMLL*XVAL;

    PLHLL = ILHLL*XVAL;
    PMHLL = IMHLL*XVAL;
    PHHLL = IHHLL*XVAL;

    PLLML = ILLML*XVAL;
    PMLML = IMLML*XVAL;
    PHLML = IHLML*XVAL;

    PLMML = ILMML*XVAL;
    PMMML = IMMML*XVAL;
    PHMML = IHMML*XVAL;

    PLHML = ILHML*XVAL;
    PMHML = IMHML*XVAL;
    PHHML = IHHML*XVAL;

    PLLHL = ILLHL*XVAL;
    PMLHL = IMLHL*XVAL;
    PHLHL = IHLHL*XVAL;

    PLMHL = ILMHL*XVAL;
    PMMHL = IMMHL*XVAL;
    PHMHL = IHMHL*XVAL;

    PLHHL = ILHHL*XVAL;
    PMHHL = IMHHL*XVAL;
    PHHHL = IHHHL*XVAL;
```

```
PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;

PLHHM = ILHHM*XVAL;
PMHHM = IMHHM*XVAL;
PHHHM = IHHHM*XVAL;

PLLLH = ILLLH*XVAL;
PMLLH = IMLLH*XVAL;
PHLLH = IHLLH*XVAL;

PLMLH = ILMLH*XVAL;
PMMLH = IMMLH*XVAL;
PHMLH = IHMLH*XVAL;

PLHLH = ILHLH*XVAL;
PMHLH = IMHLH*XVAL;
PHHLH = IHHLH*XVAL;

PLLMH = ILLMH*XVAL;
PMLMH = IMLMH*XVAL;
PHLMH = IHLMH*XVAL;
```

```
     PLMMH = ILMMH*XVAL;
     PMMMH = IMMMH*XVAL;
     PHMMH = IHMMH*XVAL;

     PLHMH = ILHMH*XVAL;
     PMHMH = IMHMH*XVAL;
     PHHMH = IHHMH*XVAL;

     PLLHH = ILLHH*XVAL;
     PMLHH = IMLHH*XVAL;
     PHLHH = IHLHH*XVAL;

     PLMHH = ILMHH*XVAL;
     PMMHH = IMMHH*XVAL;
     PHMHH = IHMHH*XVAL;

     PLHHH = ILHHH*XVAL;
     PMHHH = IMHHH*XVAL;
     PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 57: 1 or more mediators, in parallel if multiple (example uses 1), 4 moderators, 2 of which moderate both the IV-Mediator path and the direct IV-DV path with all 2-way and 3-way interactions, with the other 2 moderating both the Mediator-DV path and the direct IV-DV path with all 2-way and 3-way interactions

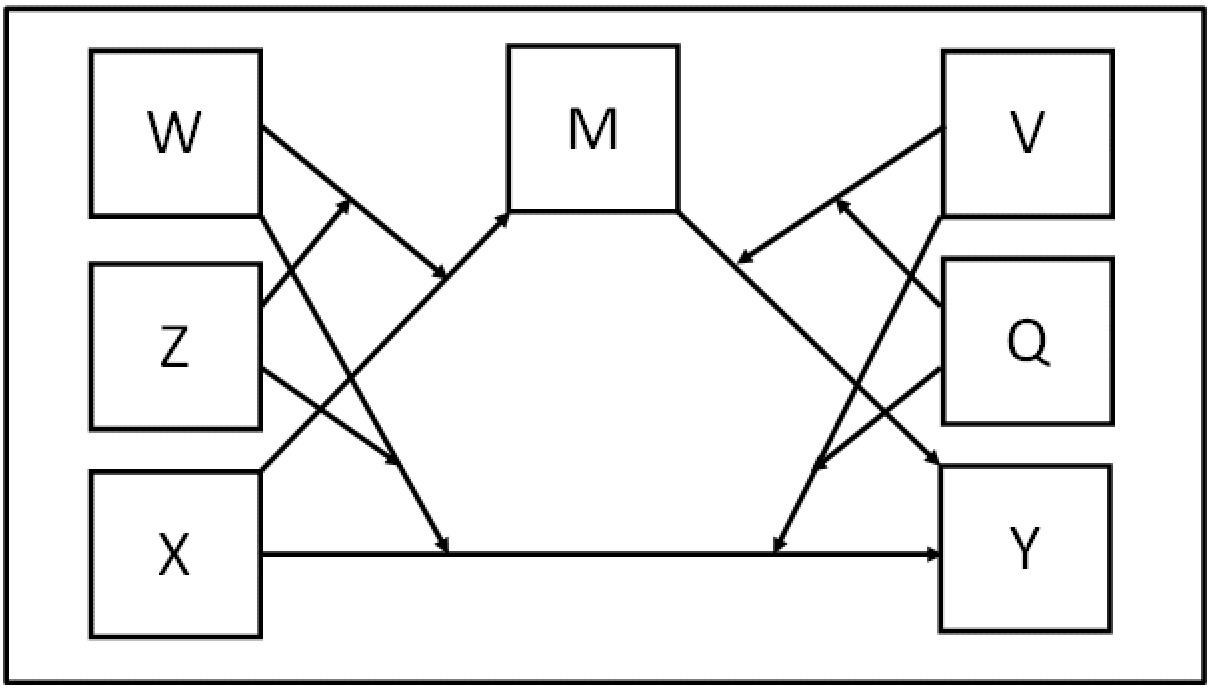
Example Variables: 1 predictor X, 1 mediator M, 4 moderators W, Z, V, Q, 1 outcome Y

Preliminary notes:

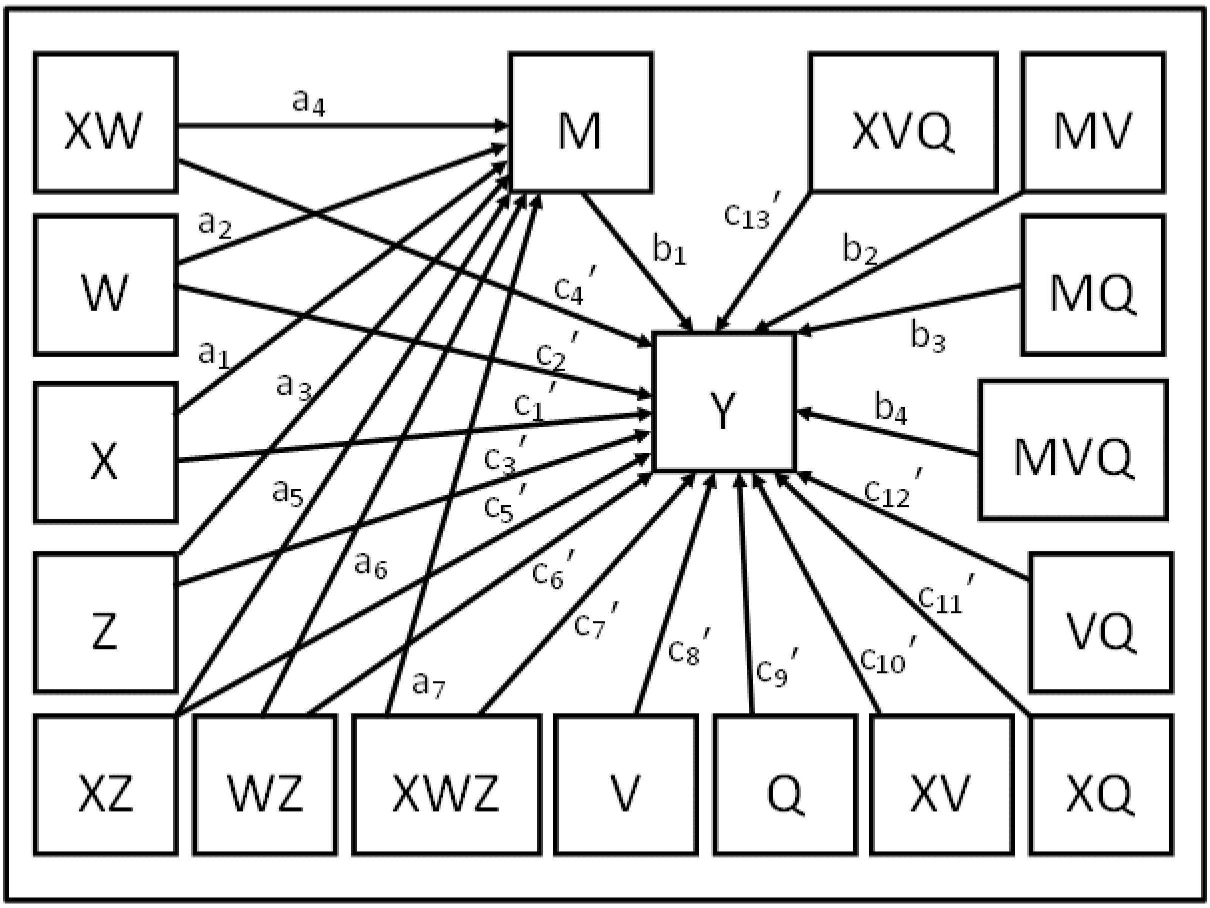
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ + c8'V + c9'Q + c10'XV + c11'XQ + c12'VQ + c13'XVQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MV + b3MQ + b4MVQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ + c8'V + c9'Q + c10'XV + c11'XQ + c12'VQ + c13'XVQ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)V + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)Q + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ +

a6WZ + a7XWZ)VQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ + c8'V + c9'Q + c10'XV + c11'XQ + c12'VQ + c13'XVQ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + a0b2V + a1b2XV + a2b2WV + a3b2ZV + a4b2XWV + a5b2XZV + a6b2WZV + a7b2XWZV + a0b3Q + a1b3XQ + a2b3WQ + a3b3ZQ + a4b3XWQ + a5b3XZQ + a6b3WZQ + a7b3XWZQ + a0b4VQ + a1b4XVQ + a2b4WVQ + a3b4ZVQ + a4b4XWVQ + a5b4XZVQ + a6b4WZVQ + a7b4XWZVQ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ + c8'V + c9'Q + c10'XV + c11'XQ + c12'VQ + c13'XVQ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + a0b2V + a2b2WV + a3b2ZV + a6b2WZV + a0b3Q + a2b3WQ + a3b3ZQ + a6b3WZQ + a0b4VQ + a2b4WVQ + a3b4ZVQ + a6b4WZVQ + c2'W + c3'Z + c6'WZ + c8'V + c9'Q + c12'VQ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV + a1b3Q + a4b3WQ + a5b3ZQ + a7b3WZQ + a1b4VQ + a4b4WVQ + a5b4ZVQ + a7b4WZVQ + c1' + c4'W + c5'Z + c7'WZ + c10'V + c11'Q + c13'VQ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z, V, Q:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2V + a4b2WV + a5b2ZV + a7b2WZV + a1b3Q + a4b3WQ + a5b3ZQ + a7b3WZQ + a1b4VQ + a4b4WVQ + a5b4ZVQ + a7b4WZVQ = (a1 + a4W + a5Z + a7WZ)(b1 + b2V + b3Q + b4VQ)

One direct effect of X on Y, conditional on W, Z, V, Q:

c1' + c4'W + c5'Z + c7'WZ + c10'V + c11'Q + c13'VQ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z, V, Q
! Outcome variable - Y

USEVARIABLES = X M W Z V Q Y XW XZ WZ XV XQ VQ MV MQ XWZ XVQ
MVQ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above
```

```
DEFINE:
   MQ = M*Q;
   MV = M*V;
   XW = X*W;
   XZ = X*Z;
   XQ = X*Q;
   XV = X*V;
   WZ = W*Z;
   VQ = V*Q;
   MVQ = M*V*Q;
   XWZ = X*W*Z;
   XVQ = X*V*Q;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MV (b2);
   Y ON MQ (b3);
   Y ON MVQ (b4);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON Z (cdash3);
   Y ON XW (cdash4);
   Y ON XZ (cdash5);
   Y ON WZ (cdash6);
   Y ON XWZ (cdash7);
   Y ON V (cdash8);
   Y ON Q (cdash9);
   Y ON XV (cdash10);
   Y ON XQ (cdash11);
   Y ON VQ (cdash12);
   Y ON XVQ (cdash13);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
```

```
    M ON WZ (a6);
    M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z, V, Q
! for example, of 1 SD below mean, mean, 1 SD above mean

! 4 moderators, 3 values for each, gives 81 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! HHML = high value of W, high value of Z, medium value of V
and low value of Q.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z LOW_V MED_V
HIGH_V LOW_Q MED_Q HIGH_Q
    ILLLL IMLLL IHLLL ILMLL IMMLL IHMLL ILHLL IMHLL IHHLL
    ILLML IMLML IHLML ILMML IMMML IHMML ILHML IMHML IHHML
    ILLHL IMLHL IHLHL ILMHL IMMHL IHMHL ILHHL IMHHL IHHHL
    ILLLM IMLLM IHLLM ILMLM IMMLM IHMLM ILHLM IMHLM IHHLM
    ILLMM IMLMM IHLMM ILMMM IMMMM IHMMM ILHMM IMHMM IHHMM
    ILLHM IMLHM IHLHM ILMHM IMMHM IHMHM ILHHM IMHHM IHHHM
    ILLLH IMLLH IHLLH ILMLH IMMLH IHMLH ILHLH IMHLH IHHLH
    ILLMH IMLMH IHLMH ILMMH IMMMH IHMMH ILHMH IMHMH IHHMH
    ILLHH IMLHH IHLHH ILMHH IMMHH IHMHH ILHHH IMHHH IHHHH
    DLLLL DMLLL DHLLL DLMLL DMMLL DHMLL DLHLL DMHLL DHHLL
    DLLML DMLML DHLML DLMML DMMML DHMML DLHML DMHML DHHML
    DLLHL DMLHL DHLHL DLMHL DMMHL DHMHL DLHHL DMHHL DHHHL
    DLLLM DMLLM DHLLM DLMLM DMMLM DHMLM DLHLM DMHLM DHHLM
    DLLMM DMLMM DHLMM DLMMM DMMMM DHMMM DLHMM DMHMM DHHMM
    DLLHM DMLHM DHLHM DLMHM DMMHM DHMHM DLHHM DMHHM DHHHM
    DLLLH DMLLH DHLLH DLMLH DMMLH DHMLH DLHLH DMHLH DHHLH
    DLLMH DMLMH DHLMH DLMMH DMMMH DHMMH DLHMH DMHMH DHHMH
    DLLHH DMLHH DHLHH DLMHH DMMHH DHMHH DLHHH DMHHH DHHHH
    TLLLL TMLLL THLLL TLMLL TMMLL THMLL TLHLL TMHLL THHLL
    TLLML TMLML THLML TLMML TMMML THMML TLHML TMHML THHML
    TLLHL TMLHL THLHL TLMHL TMMHL THMHL TLHHL TMHHL THHHL
    TLLLM TMLLM THLLM TLMLM TMMLM THMLM TLHLM TMHLM THHLM
    TLLMM TMLMM THLMM TLMMM TMMMM THMMM TLHMM TMHMM THHMM
    TLLHM TMLHM THLHM TLMHM TMMHM THMHM TLHHM TMHHM THHHM
    TLLLH TMLLH THLLH TLMLH TMMLH THMLH TLHLH TMHLH THHLH
    TLLMH TMLMH THLMH TLMMH TMMMH THMMH TLHMH TMHMH THHMH
    TLLHH TMLHH THLHH TLMHH TMMHH THMHH TLHHH TMHHH THHHH);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
```

```
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

    LOW_Q = #LOWQ;   ! replace #LOWQ in the code with your
chosen low value of Q
    MED_Q = #MEDQ;   ! replace #MEDQ in the code with your
chosen medium value of Q
    HIGH_Q = #HIGHQ;   ! replace #HIGHQ in the code with your
chosen high value of Q

! Calc conditional indirect effects for each combination of
moderator values

    ILLLL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*LOW_W*LOW_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*LOW_W*LOW_V*LOW_Q + a5*b4*LOW_Z*LOW_V*LOW_Q +
     a7*b4*LOW_W*LOW_Z*LOW_V*LOW_Q;
    IMLLL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*MED_W*LOW_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*MED_W*LOW_V*LOW_Q + a5*b4*LOW_Z*LOW_V*LOW_Q +
     a7*b4*MED_W*LOW_Z*LOW_V*LOW_Q;
    IHLLL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
```

```
     a7*b2*HIGH_W*LOW_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*HIGH_W*LOW_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*HIGH_W*LOW_V*LOW_Q + a5*b4*LOW_Z*LOW_V*LOW_Q +
     a7*b4*HIGH_W*LOW_Z*LOW_V*LOW_Q;

    ILMLL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*LOW_W*MED_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*LOW_W*LOW_V*LOW_Q + a5*b4*MED_Z*LOW_V*LOW_Q +
     a7*b4*LOW_W*MED_Z*LOW_V*LOW_Q;
    IMMLL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*MED_W*MED_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*MED_W*LOW_V*LOW_Q + a5*b4*MED_Z*LOW_V*LOW_Q +
     a7*b4*MED_W*MED_Z*LOW_V*LOW_Q;
    IHMLL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*HIGH_W*MED_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*HIGH_W*LOW_V*LOW_Q + a5*b4*MED_Z*LOW_V*LOW_Q +
     a7*b4*HIGH_W*MED_Z*LOW_V*LOW_Q;

    ILHLL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*LOW_W*HIGH_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*LOW_W*LOW_V*LOW_Q + a5*b4*HIGH_Z*LOW_V*LOW_Q +
     a7*b4*LOW_W*HIGH_Z*LOW_V*LOW_Q;
    IMHLL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V + a1*b3*LOW_Q +
```

```
a4*b3*MED_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*MED_W*HIGH_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*MED_W*LOW_V*LOW_Q + a5*b4*HIGH_Z*LOW_V*LOW_Q +
     a7*b4*MED_W*HIGH_Z*LOW_V*LOW_Q;
    IHHLL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*HIGH_W*HIGH_Z*LOW_Q +
a1*b4*LOW_V*LOW_Q +
     a4*b4*HIGH_W*LOW_V*LOW_Q + a5*b4*HIGH_Z*LOW_V*LOW_Q +
     a7*b4*HIGH_W*HIGH_Z*LOW_V*LOW_Q;

    ILLML = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*LOW_W*LOW_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*LOW_W*MED_V*LOW_Q + a5*b4*LOW_Z*MED_V*LOW_Q +
     a7*b4*LOW_W*LOW_Z*MED_V*LOW_Q;
    IMLML = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*LOW_Z*LOW_Q + a7*b3*MED_W*LOW_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*MED_W*MED_V*LOW_Q + a5*b4*LOW_Z*MED_V*LOW_Q +
     a7*b4*MED_W*LOW_Z*MED_V*LOW_Q;
    IHLML = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*HIGH_W*LOW_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*HIGH_W*MED_V*LOW_Q + a5*b4*LOW_Z*MED_V*LOW_Q +
     a7*b4*HIGH_W*LOW_Z*MED_V*LOW_Q;

    ILMML = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V + a1*b3*LOW_Q + a4*b3*LOW_W*LOW_Q
+
```

```
     a5*b3*MED_Z*LOW_Q + a7*b3*LOW_W*MED_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*LOW_W*MED_V*LOW_Q + a5*b4*MED_Z*MED_V*LOW_Q +
     a7*b4*LOW_W*MED_Z*MED_V*LOW_Q;
    IMMML = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V + a1*b3*LOW_Q + a4*b3*MED_W*LOW_Q
+
     a5*b3*MED_Z*LOW_Q + a7*b3*MED_W*MED_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*MED_W*MED_V*LOW_Q + a5*b4*MED_Z*MED_V*LOW_Q +
     a7*b4*MED_W*MED_Z*MED_V*LOW_Q;
    IHMML = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*HIGH_W*MED_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*HIGH_W*MED_V*LOW_Q + a5*b4*MED_Z*MED_V*LOW_Q +
     a7*b4*HIGH_W*MED_Z*MED_V*LOW_Q;

    ILHML = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*LOW_W*HIGH_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*LOW_W*MED_V*LOW_Q + a5*b4*HIGH_Z*MED_V*LOW_Q +
     a7*b4*LOW_W*HIGH_Z*MED_V*LOW_Q;
    IMHML = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*MED_W*HIGH_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
     a4*b4*MED_W*MED_V*LOW_Q + a5*b4*HIGH_Z*MED_V*LOW_Q +
     a7*b4*MED_W*HIGH_Z*MED_V*LOW_Q;
    IHHML = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*HIGH_W*HIGH_Z*LOW_Q +
a1*b4*MED_V*LOW_Q +
```

```
     a4*b4*HIGH_W*MED_V*LOW_Q + a5*b4*HIGH_Z*MED_V*LOW_Q +
     a7*b4*HIGH_W*HIGH_Z*MED_V*LOW_Q;

    ILLHL = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*LOW_W*LOW_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*LOW_W*HIGH_V*LOW_Q + a5*b4*LOW_Z*HIGH_V*LOW_Q +
     a7*b4*LOW_W*LOW_Z*HIGH_V*LOW_Q;
    IMLHL = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*MED_W*LOW_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*MED_W*HIGH_V*LOW_Q + a5*b4*LOW_Z*HIGH_V*LOW_Q +
     a7*b4*MED_W*LOW_Z*HIGH_V*LOW_Q;
    IHLHL = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*LOW_Z*LOW_Q + a7*b3*HIGH_W*LOW_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*HIGH_W*HIGH_V*LOW_Q + a5*b4*LOW_Z*HIGH_V*LOW_Q +
     a7*b4*HIGH_W*LOW_Z*HIGH_V*LOW_Q;

    ILMHL = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*LOW_W*MED_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*LOW_W*HIGH_V*LOW_Q + a5*b4*MED_Z*HIGH_V*LOW_Q +
     a7*b4*LOW_W*MED_Z*HIGH_V*LOW_Q;
    IMMHL = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*MED_W*MED_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
```

```
     a4*b4*MED_W*HIGH_V*LOW_Q + a5*b4*MED_Z*HIGH_V*LOW_Q +
     a7*b4*MED_W*MED_Z*HIGH_V*LOW_Q;
    IHMHL = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*MED_Z*LOW_Q + a7*b3*HIGH_W*MED_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*HIGH_W*HIGH_V*LOW_Q + a5*b4*MED_Z*HIGH_V*LOW_Q +
     a7*b4*HIGH_W*MED_Z*HIGH_V*LOW_Q;

    ILHHL = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*LOW_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*LOW_W*HIGH_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*LOW_W*HIGH_V*LOW_Q + a5*b4*HIGH_Z*HIGH_V*LOW_Q +
     a7*b4*LOW_W*HIGH_Z*HIGH_V*LOW_Q;
    IMHHL = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*MED_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*MED_W*HIGH_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*MED_W*HIGH_V*LOW_Q + a5*b4*HIGH_Z*HIGH_V*LOW_Q +
     a7*b4*MED_W*HIGH_Z*HIGH_V*LOW_Q;
    IHHHL = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V + a1*b3*LOW_Q +
a4*b3*HIGH_W*LOW_Q +
     a5*b3*HIGH_Z*LOW_Q + a7*b3*HIGH_W*HIGH_Z*LOW_Q +
a1*b4*HIGH_V*LOW_Q +
     a4*b4*HIGH_W*HIGH_V*LOW_Q + a5*b4*HIGH_Z*HIGH_V*LOW_Q +
     a7*b4*HIGH_W*HIGH_Z*HIGH_V*LOW_Q;

    ILLLM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
```

```
+
     a5*b3*LOW_Z*MED_Q + a7*b3*LOW_W*LOW_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*LOW_W*LOW_V*MED_Q + a5*b4*LOW_Z*LOW_V*MED_Q +
     a7*b4*LOW_W*LOW_Z*LOW_V*MED_Q;
    IMLLM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*MED_W*LOW_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*MED_W*LOW_V*MED_Q + a5*b4*LOW_Z*LOW_V*MED_Q +
     a7*b4*MED_W*LOW_Z*LOW_V*MED_Q;
    IHLLM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*HIGH_W*LOW_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*HIGH_W*LOW_V*MED_Q + a5*b4*LOW_Z*LOW_V*MED_Q +
     a7*b4*HIGH_W*LOW_Z*LOW_V*MED_Q;

    ILMLM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*LOW_W*MED_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*LOW_W*LOW_V*MED_Q + a5*b4*MED_Z*LOW_V*MED_Q +
     a7*b4*LOW_W*MED_Z*LOW_V*MED_Q;
    IMMLM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*MED_W*MED_Z*LOW_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*MED_W*MED_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*MED_W*LOW_V*MED_Q + a5*b4*MED_Z*LOW_V*MED_Q +
     a7*b4*MED_W*MED_Z*LOW_V*MED_Q;
    IHMLM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*HIGH_W*MED_Z*LOW_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*HIGH_W*MED_Z*MED_Q +
```

```
a1*b4*LOW_V*MED_Q +
     a4*b4*HIGH_W*LOW_V*MED_Q + a5*b4*MED_Z*LOW_V*MED_Q +
     a7*b4*HIGH_W*MED_Z*LOW_V*MED_Q;

    ILHLM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*LOW_W*HIGH_Z*LOW_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*LOW_W*HIGH_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*LOW_W*LOW_V*MED_Q + a5*b4*HIGH_Z*LOW_V*MED_Q +
     a7*b4*LOW_W*HIGH_Z*LOW_V*MED_Q;
    IMHLM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*MED_W*HIGH_Z*LOW_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*MED_W*HIGH_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*MED_W*LOW_V*MED_Q + a5*b4*HIGH_Z*LOW_V*MED_Q +
     a7*b4*MED_W*HIGH_Z*LOW_V*MED_Q;
    IHHLM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
     a7*b2*HIGH_W*HIGH_Z*LOW_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*HIGH_W*HIGH_Z*MED_Q +
a1*b4*LOW_V*MED_Q +
     a4*b4*HIGH_W*LOW_V*MED_Q + a5*b4*HIGH_Z*LOW_V*MED_Q +
     a7*b4*HIGH_W*HIGH_Z*LOW_V*MED_Q;

    ILLMM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*LOW_W*LOW_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*LOW_W*MED_V*MED_Q + a5*b4*LOW_Z*MED_V*MED_Q +
     a7*b4*LOW_W*LOW_Z*MED_V*MED_Q;
    IMLMM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*LOW_Z*MED_Q + a7*b3*MED_W*LOW_Z*MED_Q +
a1*b4*MED_V*MED_Q +
```

```
     a4*b4*MED_W*MED_V*MED_Q + a5*b4*LOW_Z*MED_V*MED_Q +
     a7*b4*MED_W*LOW_Z*MED_V*MED_Q;
    IHLMM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*HIGH_W*LOW_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*HIGH_W*MED_V*MED_Q + a5*b4*LOW_Z*MED_V*MED_Q +
     a7*b4*HIGH_W*LOW_Z*MED_V*MED_Q;

    ILMMM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V + a1*b3*MED_Q + a4*b3*LOW_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*LOW_W*MED_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*LOW_W*MED_V*MED_Q + a5*b4*MED_Z*MED_V*MED_Q +
     a7*b4*LOW_W*MED_Z*MED_V*MED_Q;
    IMMMM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V + a1*b3*MED_Q + a4*b3*MED_W*MED_Q
+
     a5*b3*MED_Z*MED_Q + a7*b3*MED_W*MED_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*MED_W*MED_V*MED_Q + a5*b4*MED_Z*MED_V*MED_Q +
     a7*b4*MED_W*MED_Z*MED_V*MED_Q;
    IHMMM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*HIGH_W*MED_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*HIGH_W*MED_V*MED_Q + a5*b4*MED_Z*MED_V*MED_Q +
     a7*b4*HIGH_W*MED_Z*MED_V*MED_Q;

    ILHMM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*LOW_W*HIGH_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*LOW_W*MED_V*MED_Q + a5*b4*HIGH_Z*MED_V*MED_Q +
```

```
     a7*b4*LOW_W*HIGH_Z*MED_V*MED_Q;
    IMHMM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*MED_W*HIGH_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*MED_W*MED_V*MED_Q + a5*b4*HIGH_Z*MED_V*MED_Q +
     a7*b4*MED_W*HIGH_Z*MED_V*MED_Q;
    IHHMM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*HIGH_W*HIGH_Z*MED_Q +
a1*b4*MED_V*MED_Q +
     a4*b4*HIGH_W*MED_V*MED_Q + a5*b4*HIGH_Z*MED_V*MED_Q +
     a7*b4*HIGH_W*HIGH_Z*MED_V*MED_Q;

    ILLHM = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*LOW_W*LOW_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*LOW_W*HIGH_V*MED_Q + a5*b4*LOW_Z*HIGH_V*MED_Q +
     a7*b4*LOW_W*LOW_Z*HIGH_V*MED_Q;
    IMLHM = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*MED_W*LOW_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*MED_W*LOW_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*MED_W*HIGH_V*MED_Q + a5*b4*LOW_Z*HIGH_V*MED_Q +
     a7*b4*MED_W*LOW_Z*HIGH_V*MED_Q;
    IHLHM = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*LOW_Z*MED_Q + a7*b3*HIGH_W*LOW_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*HIGH_W*HIGH_V*MED_Q + a5*b4*LOW_Z*HIGH_V*MED_Q +
     a7*b4*HIGH_W*LOW_Z*HIGH_V*MED_Q;
```

```
    ILMHM = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*LOW_W*MED_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*LOW_W*HIGH_V*MED_Q + a5*b4*MED_Z*HIGH_V*MED_Q +
     a7*b4*LOW_W*MED_Z*HIGH_V*MED_Q;
    IMMHM = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*MED_W*MED_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*MED_W*HIGH_V*MED_Q + a5*b4*MED_Z*HIGH_V*MED_Q +
     a7*b4*MED_W*MED_Z*HIGH_V*MED_Q;
    IHMHM = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*MED_Z*MED_Q + a7*b3*HIGH_W*MED_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*HIGH_W*HIGH_V*MED_Q + a5*b4*MED_Z*HIGH_V*MED_Q +
     a7*b4*HIGH_W*MED_Z*HIGH_V*MED_Q;

    ILHHM = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*LOW_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*LOW_W*HIGH_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*LOW_W*HIGH_V*MED_Q + a5*b4*HIGH_Z*HIGH_V*MED_Q +
     a7*b4*LOW_W*HIGH_Z*HIGH_V*MED_Q;
    IMHHM = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*MED_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*MED_W*HIGH_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*MED_W*HIGH_V*MED_Q + a5*b4*HIGH_Z*HIGH_V*MED_Q +
```

```
     a7*b4*MED_W*HIGH_Z*HIGH_V*MED_Q;
    IHHHM = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V + a1*b3*MED_Q +
a4*b3*HIGH_W*MED_Q +
     a5*b3*HIGH_Z*MED_Q + a7*b3*HIGH_W*HIGH_Z*MED_Q +
a1*b4*HIGH_V*MED_Q +
     a4*b4*HIGH_W*HIGH_V*MED_Q + a5*b4*HIGH_Z*HIGH_V*MED_Q +
     a7*b4*HIGH_W*HIGH_Z*HIGH_V*MED_Q;

    ILLLH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*LOW_W*LOW_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*LOW_W*LOW_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*LOW_W*LOW_V*HIGH_Q + a5*b4*LOW_Z*LOW_V*HIGH_Q +
     a7*b4*LOW_W*LOW_Z*LOW_V*HIGH_Q;
    IMLLH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*MED_W*LOW_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*MED_W*LOW_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*MED_W*LOW_V*HIGH_Q + a5*b4*LOW_Z*LOW_V*HIGH_Q +
     a7*b4*MED_W*LOW_Z*LOW_V*HIGH_Q;
    IHLLH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*LOW_Z*LOW_V +
     a7*b2*HIGH_W*LOW_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*HIGH_W*LOW_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*HIGH_W*LOW_V*HIGH_Q + a5*b4*LOW_Z*LOW_V*HIGH_Q +
     a7*b4*HIGH_W*LOW_Z*LOW_V*HIGH_Q;

    ILMLH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*MED_Z*LOW_V +
     a7*b2*LOW_W*MED_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*LOW_W*MED_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
     a4*b4*LOW_W*LOW_V*HIGH_Q + a5*b4*MED_Z*LOW_V*HIGH_Q +
```

```
      a7*b4*LOW_W*MED_Z*LOW_V*HIGH_Q;
    IMMLH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
      a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*MED_Z*LOW_V +
      a7*b2*MED_W*MED_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
      a5*b3*MED_Z*HIGH_Q + a7*b3*MED_W*MED_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
      a4*b4*MED_W*LOW_V*HIGH_Q + a5*b4*MED_Z*LOW_V*HIGH_Q +
      a7*b4*MED_W*MED_Z*LOW_V*HIGH_Q;
    IHMLH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
      a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*MED_Z*LOW_V +
      a7*b2*HIGH_W*MED_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
      a5*b3*MED_Z*HIGH_Q + a7*b3*HIGH_W*MED_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
      a4*b4*HIGH_W*LOW_V*HIGH_Q + a5*b4*MED_Z*LOW_V*HIGH_Q +
      a7*b4*HIGH_W*MED_Z*LOW_V*HIGH_Q;

    ILHLH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
      a1*b2*LOW_V + a4*b2*LOW_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
      a7*b2*LOW_W*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
      a5*b3*HIGH_Z*HIGH_Q + a7*b3*LOW_W*HIGH_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
      a4*b4*LOW_W*LOW_V*HIGH_Q + a5*b4*HIGH_Z*LOW_V*HIGH_Q +
      a7*b4*LOW_W*HIGH_Z*LOW_V*HIGH_Q;
    IMHLH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
      a1*b2*LOW_V + a4*b2*MED_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
      a7*b2*MED_W*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
      a5*b3*HIGH_Z*HIGH_Q + a7*b3*MED_W*HIGH_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
      a4*b4*MED_W*LOW_V*HIGH_Q + a5*b4*HIGH_Z*LOW_V*HIGH_Q +
      a7*b4*MED_W*HIGH_Z*LOW_V*HIGH_Q;
    IHHLH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
      a1*b2*LOW_V + a4*b2*HIGH_W*LOW_V + a5*b2*HIGH_Z*LOW_V +
      a7*b2*HIGH_W*HIGH_Z*LOW_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
      a5*b3*HIGH_Z*HIGH_Q + a7*b3*HIGH_W*HIGH_Z*HIGH_Q +
a1*b4*LOW_V*HIGH_Q +
      a4*b4*HIGH_W*LOW_V*HIGH_Q + a5*b4*HIGH_Z*LOW_V*HIGH_Q +
      a7*b4*HIGH_W*HIGH_Z*LOW_V*HIGH_Q;
```

```
    ILLMH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*LOW_W*LOW_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*LOW_W*LOW_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*LOW_W*MED_V*HIGH_Q + a5*b4*LOW_Z*MED_V*HIGH_Q +
     a7*b4*LOW_W*LOW_Z*MED_V*HIGH_Q;
    IMLMH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*MED_W*LOW_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*MED_W*LOW_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*MED_W*MED_V*HIGH_Q + a5*b4*LOW_Z*MED_V*HIGH_Q +
     a7*b4*MED_W*LOW_Z*MED_V*HIGH_Q;
    IHLMH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*LOW_Z*MED_V +
     a7*b2*HIGH_W*LOW_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*HIGH_W*LOW_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*HIGH_W*MED_V*HIGH_Q + a5*b4*LOW_Z*MED_V*HIGH_Q +
     a7*b4*HIGH_W*LOW_Z*MED_V*HIGH_Q;

    ILMMH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*LOW_W*MED_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*LOW_W*MED_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*LOW_W*MED_V*HIGH_Q + a5*b4*MED_Z*MED_V*HIGH_Q +
     a7*b4*LOW_W*MED_Z*MED_V*HIGH_Q;
    IMMMH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*MED_W*MED_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*MED_W*MED_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*MED_W*MED_V*HIGH_Q + a5*b4*MED_Z*MED_V*HIGH_Q +
     a7*b4*MED_W*MED_Z*MED_V*HIGH_Q;
    IHMMH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
```

```
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*MED_Z*MED_V +
     a7*b2*HIGH_W*MED_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*HIGH_W*MED_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*HIGH_W*MED_V*HIGH_Q + a5*b4*MED_Z*MED_V*HIGH_Q +
     a7*b4*HIGH_W*MED_Z*MED_V*HIGH_Q;

    ILHMH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*LOW_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*LOW_W*HIGH_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*LOW_W*HIGH_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*LOW_W*MED_V*HIGH_Q + a5*b4*HIGH_Z*MED_V*HIGH_Q +
     a7*b4*LOW_W*HIGH_Z*MED_V*HIGH_Q;
    IMHMH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*MED_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*MED_W*HIGH_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*MED_W*HIGH_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*MED_W*MED_V*HIGH_Q + a5*b4*HIGH_Z*MED_V*HIGH_Q +
     a7*b4*MED_W*HIGH_Z*MED_V*HIGH_Q;
    IHHMH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*MED_V + a4*b2*HIGH_W*MED_V + a5*b2*HIGH_Z*MED_V +
     a7*b2*HIGH_W*HIGH_Z*MED_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*HIGH_W*HIGH_Z*HIGH_Q +
a1*b4*MED_V*HIGH_Q +
     a4*b4*HIGH_W*MED_V*HIGH_Q + a5*b4*HIGH_Z*MED_V*HIGH_Q +
     a7*b4*HIGH_W*HIGH_Z*MED_V*HIGH_Q;

    ILLHH = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
     a7*b2*LOW_W*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*LOW_W*LOW_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*LOW_W*HIGH_V*HIGH_Q + a5*b4*LOW_Z*HIGH_V*HIGH_Q +
     a7*b4*LOW_W*LOW_Z*HIGH_V*HIGH_Q;
    IMLHH = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*LOW_Z*HIGH_V +
```

```
     a7*b2*MED_W*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*MED_W*LOW_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*MED_W*HIGH_V*HIGH_Q + a5*b4*LOW_Z*HIGH_V*HIGH_Q +
     a7*b4*MED_W*LOW_Z*HIGH_V*HIGH_Q;
    IHLHH = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*LOW_Z*HIGH_V
+
     a7*b2*HIGH_W*LOW_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*LOW_Z*HIGH_Q + a7*b3*HIGH_W*LOW_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*HIGH_W*HIGH_V*HIGH_Q + a5*b4*LOW_Z*HIGH_V*HIGH_Q +
     a7*b4*HIGH_W*LOW_Z*HIGH_V*HIGH_Q;

    ILMHH = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*LOW_W*MED_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*LOW_W*MED_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*LOW_W*HIGH_V*HIGH_Q + a5*b4*MED_Z*HIGH_V*HIGH_Q +
     a7*b4*LOW_W*MED_Z*HIGH_V*HIGH_Q;
    IMMHH = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*MED_Z*HIGH_V +
     a7*b2*MED_W*MED_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*MED_W*MED_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*MED_W*HIGH_V*HIGH_Q + a5*b4*MED_Z*HIGH_V*HIGH_Q +
     a7*b4*MED_W*MED_Z*HIGH_V*HIGH_Q;
    IHMHH = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*MED_Z*HIGH_V
+
     a7*b2*HIGH_W*MED_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*MED_Z*HIGH_Q + a7*b3*HIGH_W*MED_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*HIGH_W*HIGH_V*HIGH_Q + a5*b4*MED_Z*HIGH_V*HIGH_Q +
     a7*b4*HIGH_W*MED_Z*HIGH_V*HIGH_Q;

    ILHHH = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
```

```
     a1*b2*HIGH_V + a4*b2*LOW_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*LOW_W*HIGH_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*LOW_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*LOW_W*HIGH_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*LOW_W*HIGH_V*HIGH_Q + a5*b4*HIGH_Z*HIGH_V*HIGH_Q +
     a7*b4*LOW_W*HIGH_Z*HIGH_V*HIGH_Q;
    IMHHH = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*MED_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*MED_W*HIGH_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*MED_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*MED_W*HIGH_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*MED_W*HIGH_V*HIGH_Q + a5*b4*HIGH_Z*HIGH_V*HIGH_Q +
     a7*b4*MED_W*HIGH_Z*HIGH_V*HIGH_Q;
    IHHHH = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_V + a4*b2*HIGH_W*HIGH_V + a5*b2*HIGH_Z*HIGH_V
+
     a7*b2*HIGH_W*HIGH_Z*HIGH_V + a1*b3*HIGH_Q +
a4*b3*HIGH_W*HIGH_Q +
     a5*b3*HIGH_Z*HIGH_Q + a7*b3*HIGH_W*HIGH_Z*HIGH_Q +
a1*b4*HIGH_V*HIGH_Q +
     a4*b4*HIGH_W*HIGH_V*HIGH_Q + a5*b4*HIGH_Z*HIGH_V*HIGH_Q
+
     a7*b4*HIGH_W*HIGH_Z*HIGH_V*HIGH_Q;

! Calc conditional direct effects for each combination of
moderator values

    DLLLL = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash10*LOW_V + cdash11*LOW_Q + cdash13*LOW_V*LOW_Q;
    DMLLL = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash10*LOW_V + cdash11*LOW_Q + cdash13*LOW_V*LOW_Q;
    DHLLL = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash10*LOW_V + cdash11*LOW_Q + cdash13*LOW_V*LOW_Q;

    DLMLL = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash10*LOW_V + cdash11*LOW_Q + cdash13*LOW_V*LOW_Q;
    DMMLL = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash10*LOW_V + cdash11*LOW_Q + cdash13*LOW_V*LOW_Q;
```

```
    DHMLL = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash10*LOW_V + cdash11*LOW_Q + cdash13*LOW_V*LOW_Q;

    DLHLL = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash10*LOW_V + cdash11*LOW_Q + cdash13*LOW_V*LOW_Q;
    DMHLL = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash10*LOW_V + cdash11*LOW_Q + cdash13*LOW_V*LOW_Q;
    DHHLL = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash10*LOW_V + cdash11*LOW_Q + cdash13*LOW_V*LOW_Q;

    DLLML = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash10*MED_V + cdash11*LOW_Q + cdash13*MED_V*LOW_Q;
    DMLML = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash10*MED_V + cdash11*LOW_Q + cdash13*MED_V*LOW_Q;
    DHLML = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash10*MED_V + cdash11*LOW_Q + cdash13*MED_V*LOW_Q;

    DLMML = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash10*MED_V + cdash11*LOW_Q + cdash13*MED_V*LOW_Q;
    DMMML = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash10*MED_V + cdash11*LOW_Q + cdash13*MED_V*LOW_Q;
    DHMML = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash10*MED_V + cdash11*LOW_Q + cdash13*MED_V*LOW_Q;

    DLHML = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash10*MED_V + cdash11*LOW_Q + cdash13*MED_V*LOW_Q;
    DMHML = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash10*MED_V + cdash11*LOW_Q + cdash13*MED_V*LOW_Q;
    DHHML = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash10*MED_V + cdash11*LOW_Q + cdash13*MED_V*LOW_Q;

    DLLHL = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash10*HIGH_V + cdash11*LOW_Q + cdash13*HIGH_V*LOW_Q;
    DMLHL = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
```

```
     cdash10*HIGH_V + cdash11*LOW_Q + cdash13*HIGH_V*LOW_Q;
    DHLHL = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash10*HIGH_V + cdash11*LOW_Q + cdash13*HIGH_V*LOW_Q;

    DLMHL = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash10*HIGH_V + cdash11*LOW_Q + cdash13*HIGH_V*LOW_Q;
    DMMHL = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash10*HIGH_V + cdash11*LOW_Q + cdash13*HIGH_V*LOW_Q;
    DHMHL = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash10*HIGH_V + cdash11*LOW_Q + cdash13*HIGH_V*LOW_Q;

    DLHHL = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash10*HIGH_V + cdash11*LOW_Q + cdash13*HIGH_V*LOW_Q;
    DMHHL = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash10*HIGH_V + cdash11*LOW_Q + cdash13*HIGH_V*LOW_Q;
    DHHHL = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash10*HIGH_V + cdash11*LOW_Q + cdash13*HIGH_V*LOW_Q;

    DLLLM = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash10*LOW_V + cdash11*MED_Q + cdash13*LOW_V*MED_Q;
    DMLLM = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash10*LOW_V + cdash11*MED_Q + cdash13*LOW_V*MED_Q;
    DHLLM = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash10*LOW_V + cdash11*MED_Q + cdash13*LOW_V*MED_Q;

    DLMLM = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash10*LOW_V + cdash11*MED_Q + cdash13*LOW_V*MED_Q;
    DMMLM = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash10*LOW_V + cdash11*MED_Q + cdash13*LOW_V*MED_Q;
    DHMLM = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash10*LOW_V + cdash11*MED_Q + cdash13*LOW_V*MED_Q;

    DLHLM = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash10*LOW_V + cdash11*MED_Q + cdash13*LOW_V*MED_Q;
    DMHLM = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
```

```
cdash7*MED_W*HIGH_Z +
     cdash10*LOW_V + cdash11*MED_Q + cdash13*LOW_V*MED_Q;
    DHHLM = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash10*LOW_V + cdash11*MED_Q + cdash13*LOW_V*MED_Q;

    DLLMM = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash10*MED_V + cdash11*MED_Q + cdash13*MED_V*MED_Q;
    DMLMM = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash10*MED_V + cdash11*MED_Q + cdash13*MED_V*MED_Q;
    DHLMM = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash10*MED_V + cdash11*MED_Q + cdash13*MED_V*MED_Q;

    DLMMM = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash10*MED_V + cdash11*MED_Q + cdash13*MED_V*MED_Q;
    DMMMM = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash10*MED_V + cdash11*MED_Q + cdash13*MED_V*MED_Q;
    DHMMM = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash10*MED_V + cdash11*MED_Q + cdash13*MED_V*MED_Q;

    DLHMM = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash10*MED_V + cdash11*MED_Q + cdash13*MED_V*MED_Q;
    DMHMM = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash10*MED_V + cdash11*MED_Q + cdash13*MED_V*MED_Q;
    DHHMM = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash10*MED_V + cdash11*MED_Q + cdash13*MED_V*MED_Q;

    DLLHM = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash10*HIGH_V + cdash11*MED_Q + cdash13*HIGH_V*MED_Q;
    DMLHM = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash10*HIGH_V + cdash11*MED_Q + cdash13*HIGH_V*MED_Q;
    DHLHM = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash10*HIGH_V + cdash11*MED_Q + cdash13*HIGH_V*MED_Q;

    DLMHM = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash10*HIGH_V + cdash11*MED_Q + cdash13*HIGH_V*MED_Q;
```

```
    DMMHM = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash10*HIGH_V + cdash11*MED_Q + cdash13*HIGH_V*MED_Q;
    DHMHM = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash10*HIGH_V + cdash11*MED_Q + cdash13*HIGH_V*MED_Q;

    DLHHM = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash10*HIGH_V + cdash11*MED_Q + cdash13*HIGH_V*MED_Q;
    DMHHM = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash10*HIGH_V + cdash11*MED_Q + cdash13*HIGH_V*MED_Q;
    DHHHM = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash10*HIGH_V + cdash11*MED_Q + cdash13*HIGH_V*MED_Q;

    DLLLH = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash10*LOW_V + cdash11*HIGH_Q + cdash13*LOW_V*HIGH_Q;
    DMLLH = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash10*LOW_V + cdash11*HIGH_Q + cdash13*LOW_V*HIGH_Q;
    DHLLH = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash10*LOW_V + cdash11*HIGH_Q + cdash13*LOW_V*HIGH_Q;

    DLMLH = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash10*LOW_V + cdash11*HIGH_Q + cdash13*LOW_V*HIGH_Q;
    DMMLH = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash10*LOW_V + cdash11*HIGH_Q + cdash13*LOW_V*HIGH_Q;
    DHMLH = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash10*LOW_V + cdash11*HIGH_Q + cdash13*LOW_V*HIGH_Q;

    DLHLH = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash10*LOW_V + cdash11*HIGH_Q + cdash13*LOW_V*HIGH_Q;
    DMHLH = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash10*LOW_V + cdash11*HIGH_Q + cdash13*LOW_V*HIGH_Q;
    DHHLH = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash10*LOW_V + cdash11*HIGH_Q + cdash13*LOW_V*HIGH_Q;

    DLLMH = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
```

```
     cdash10*MED_V + cdash11*HIGH_Q + cdash13*MED_V*HIGH_Q;
    DMLMH = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash10*MED_V + cdash11*HIGH_Q + cdash13*MED_V*HIGH_Q;
    DHLMH = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash10*MED_V + cdash11*HIGH_Q + cdash13*MED_V*HIGH_Q;

    DLMMH = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash10*MED_V + cdash11*HIGH_Q + cdash13*MED_V*HIGH_Q;
    DMMMH = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash10*MED_V + cdash11*HIGH_Q + cdash13*MED_V*HIGH_Q;
    DHMMH = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash10*MED_V + cdash11*HIGH_Q + cdash13*MED_V*HIGH_Q;

    DLHMH = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash10*MED_V + cdash11*HIGH_Q + cdash13*MED_V*HIGH_Q;
    DMHMH = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash10*MED_V + cdash11*HIGH_Q + cdash13*MED_V*HIGH_Q;
    DHHMH = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash10*MED_V + cdash11*HIGH_Q + cdash13*MED_V*HIGH_Q;

    DLLHH = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z +
     cdash10*HIGH_V + cdash11*HIGH_Q + cdash13*HIGH_V*HIGH_Q;
    DMLHH = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z +
     cdash10*HIGH_V + cdash11*HIGH_Q + cdash13*HIGH_V*HIGH_Q;
    DHLHH = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z +
     cdash10*HIGH_V + cdash11*HIGH_Q + cdash13*HIGH_V*HIGH_Q;

    DLMHH = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z +
     cdash10*HIGH_V + cdash11*HIGH_Q + cdash13*HIGH_V*HIGH_Q;
    DMMHH = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z +
     cdash10*HIGH_V + cdash11*HIGH_Q + cdash13*HIGH_V*HIGH_Q;
    DHMHH = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z +
     cdash10*HIGH_V + cdash11*HIGH_Q + cdash13*HIGH_V*HIGH_Q;
```

```
    DLHHH = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z +
     cdash10*HIGH_V + cdash11*HIGH_Q + cdash13*HIGH_V*HIGH_Q;
    DMHHH = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z +
     cdash10*HIGH_V + cdash11*HIGH_Q + cdash13*HIGH_V*HIGH_Q;
    DHHHH = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z +
     cdash10*HIGH_V + cdash11*HIGH_Q + cdash13*HIGH_V*HIGH_Q;

! Calc conditional total effects for each combination of
moderator values

    TLLLL = ILLLL + DLLLL;
    TMLLL = IMLLL + DMLLL;
    THLLL = IHLLL + DHLLL;

    TLMLL = ILMLL + DLMLL;
    TMMLL = IMMLL + DMMLL;
    THMLL = IHMLL + DHMLL;

    TLHLL = ILHLL + DLHLL;
    TMHLL = IMHLL + DMHLL;
    THHLL = IHHLL + DHHLL;

    TLLML = ILLML + DLLML;
    TMLML = IMLML + DMLML;
    THLML = IHLML + DHLML;

    TLMML = ILMML + DLMML;
    TMMML = IMMML + DMMML;
    THMML = IHMML + DHMML;

    TLHML = ILHML + DLHML;
    TMHML = IMHML + DMHML;
    THHML = IHHML + DHHML;

    TLLHL = ILLHL + DLLHL;
    TMLHL = IMLHL + DMLHL;
    THLHL = IHLHL + DHLHL;

    TLMHL = ILMHL + DLMHL;
    TMMHL = IMMHL + DMMHL;
    THMHL = IHMHL + DHMHL;

    TLHHL = ILHHL + DLHHL;
    TMHHL = IMHHL + DMHHL;
    THHHL = IHHHL + DHHHL;

    TLLLM = ILLLM + DLLLM;
    TMLLM = IMLLM + DMLLM;
    THLLM = IHLLM + DHLLM;
```

```
TLMLM = ILMLM + DLMLM;
TMMLM = IMMLM + DMMLM;
THMLM = IHMLM + DHMLM;

TLHLM = ILHLM + DLHLM;
TMHLM = IMHLM + DMHLM;
THHLM = IHHLM + DHHLM;

TLLMM = ILLMM + DLLMM;
TMLMM = IMLMM + DMLMM;
THLMM = IHLMM + DHLMM;

TLMMM = ILMMM + DLMMM;
TMMMM = IMMMM + DMMMM;
THMMM = IHMMM + DHMMM;

TLHMM = ILHMM + DLHMM;
TMHMM = IMHMM + DMHMM;
THHMM = IHHMM + DHHMM;

TLLHM = ILLHM + DLLHM;
TMLHM = IMLHM + DMLHM;
THLHM = IHLHM + DHLHM;

TLMHM = ILMHM + DLMHM;
TMMHM = IMMHM + DMMHM;
THMHM = IHMHM + DHMHM;

TLHHM = ILHHM + DLHHM;
TMHHM = IMHHM + DMHHM;
THHHM = IHHHM + DHHHM;

TLLLH = ILLLH + DLLLH;
TMLLH = IMLLH + DMLLH;
THLLH = IHLLH + DHLLH;

TLMLH = ILMLH + DLMLH;
TMMLH = IMMLH + DMMLH;
THMLH = IHMLH + DHMLH;

TLHLH = ILHLH + DLHLH;
TMHLH = IMHLH + DMHLH;
THHLH = IHHLH + DHHLH;

TLLMH = ILLMH + DLLMH;
TMLMH = IMLMH + DMLMH;
THLMH = IHLMH + DHLMH;

TLMMH = ILMMH + DLMMH;
TMMMH = IMMMH + DMMMH;
THMMH = IHMMH + DHMMH;
```

```
    TLHMH = ILHMH + DLHMH;
    TMHMH = IMHMH + DMHMH;
    THHMH = IHHMH + DHHMH;

    TLLHH = ILLHH + DLLHH;
    TMLHH = IMLHH + DMLHH;
    THLHH = IHLHH + DHLHH;

    TLMHH = ILMHH + DLMHH;
    TMMHH = IMMHH + DMMHH;
    THMHH = IHMHH + DHMHH;

    TLHHH = ILHHH + DLHHH;
    TMHHH = IMHHH + DMHHH;
    THHHH = IHHHH + DHHHH;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLLLL PMLLL PHLLL PLMLL PMMLL PHMLL PLHLL PMHLL
PHHLL
    PLLML PMLML PHLML PLMML PMMML PHMML PLHML PMHML PHHML
    PLLHL PMLHL PHLHL PLMHL PMMHL PHMHL PLHHL PMHHL PHHHL
    PLLLM PMLLM PHLLM PLMLM PMMLM PHMLM PLHLM PMHLM PHHLM
    PLLMM PMLMM PHLMM PLMMM PMMMM PHMMM PLHMM PMHMM PHHMM
    PLLHM PMLHM PHLHM PLMHM PMMHM PHMHM PLHHM PMHHM PHHHM
    PLLLH PMLLH PHLLH PLMLH PMMLH PHMLH PLHLH PMHLH PHHLH
    PLLMH PMLMH PHLMH PLMMH PMMMH PHMMH PLHMH PMHMH PHHMH
    PLLHH PMLHH PHLHH PLMHH PMMHH PHMHH PLHHH PMHHH PHHHH);

    LOOP(XVAL,1,5,0.1);

    PLLLL = ILLLL*XVAL;
    PMLLL = IMLLL*XVAL;
    PHLLL = IHLLL*XVAL;

    PLMLL = ILMLL*XVAL;
    PMMLL = IMMLL*XVAL;
    PHMLL = IHMLL*XVAL;

    PLHLL = ILHLL*XVAL;
    PMHLL = IMHLL*XVAL;
    PHHLL = IHHLL*XVAL;

    PLLML = ILLML*XVAL;
    PMLML = IMLML*XVAL;
    PHLML = IHLML*XVAL;
```

```
PLMML = ILMML*XVAL;
PMMML = IMMML*XVAL;
PHMML = IHMML*XVAL;

PLHML = ILHML*XVAL;
PMHML = IMHML*XVAL;
PHHML = IHHML*XVAL;

PLLHL = ILLHL*XVAL;
PMLHL = IMLHL*XVAL;
PHLHL = IHLHL*XVAL;

PLMHL = ILMHL*XVAL;
PMMHL = IMMHL*XVAL;
PHMHL = IHMHL*XVAL;

PLHHL = ILHHL*XVAL;
PMHHL = IMHHL*XVAL;
PHHHL = IHHHL*XVAL;

PLLLM = ILLLM*XVAL;
PMLLM = IMLLM*XVAL;
PHLLM = IHLLM*XVAL;

PLMLM = ILMLM*XVAL;
PMMLM = IMMLM*XVAL;
PHMLM = IHMLM*XVAL;

PLHLM = ILHLM*XVAL;
PMHLM = IMHLM*XVAL;
PHHLM = IHHLM*XVAL;

PLLMM = ILLMM*XVAL;
PMLMM = IMLMM*XVAL;
PHLMM = IHLMM*XVAL;

PLMMM = ILMMM*XVAL;
PMMMM = IMMMM*XVAL;
PHMMM = IHMMM*XVAL;

PLHMM = ILHMM*XVAL;
PMHMM = IMHMM*XVAL;
PHHMM = IHHMM*XVAL;

PLLHM = ILLHM*XVAL;
PMLHM = IMLHM*XVAL;
PHLHM = IHLHM*XVAL;

PLMHM = ILMHM*XVAL;
PMMHM = IMMHM*XVAL;
PHMHM = IHMHM*XVAL;
```

```
     PLHHM = ILHHM*XVAL;
     PMHHM = IMHHM*XVAL;
     PHHHM = IHHHM*XVAL;

     PLLLH = ILLLH*XVAL;
     PMLLH = IMLLH*XVAL;
     PHLLH = IHLLH*XVAL;

     PLMLH = ILMLH*XVAL;
     PMMLH = IMMLH*XVAL;
     PHMLH = IHMLH*XVAL;

     PLHLH = ILHLH*XVAL;
     PMHLH = IMHLH*XVAL;
     PHHLH = IHHLH*XVAL;

     PLLMH = ILLMH*XVAL;
     PMLMH = IMLMH*XVAL;
     PHLMH = IHLMH*XVAL;

     PLMMH = ILMMH*XVAL;
     PMMMH = IMMMH*XVAL;
     PHMMH = IHMMH*XVAL;

     PLHMH = ILHMH*XVAL;
     PMHMH = IMHMH*XVAL;
     PHHMH = IHHMH*XVAL;

     PLLHH = ILLHH*XVAL;
     PMLHH = IMLHH*XVAL;
     PHLHH = IHLHH*XVAL;

     PLMHH = ILMHH*XVAL;
     PMMHH = IMMHH*XVAL;
     PHMHH = IHMHH*XVAL;

     PLHHH = ILHHH*XVAL;
     PMHHH = IMHHH*XVAL;
     PHHHH = IHHHH*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 58: 1 or more mediators, in parallel if multiple (example uses 1), 1 moderators, which moderates both the IV- Mediator path and the Mediator-DV path

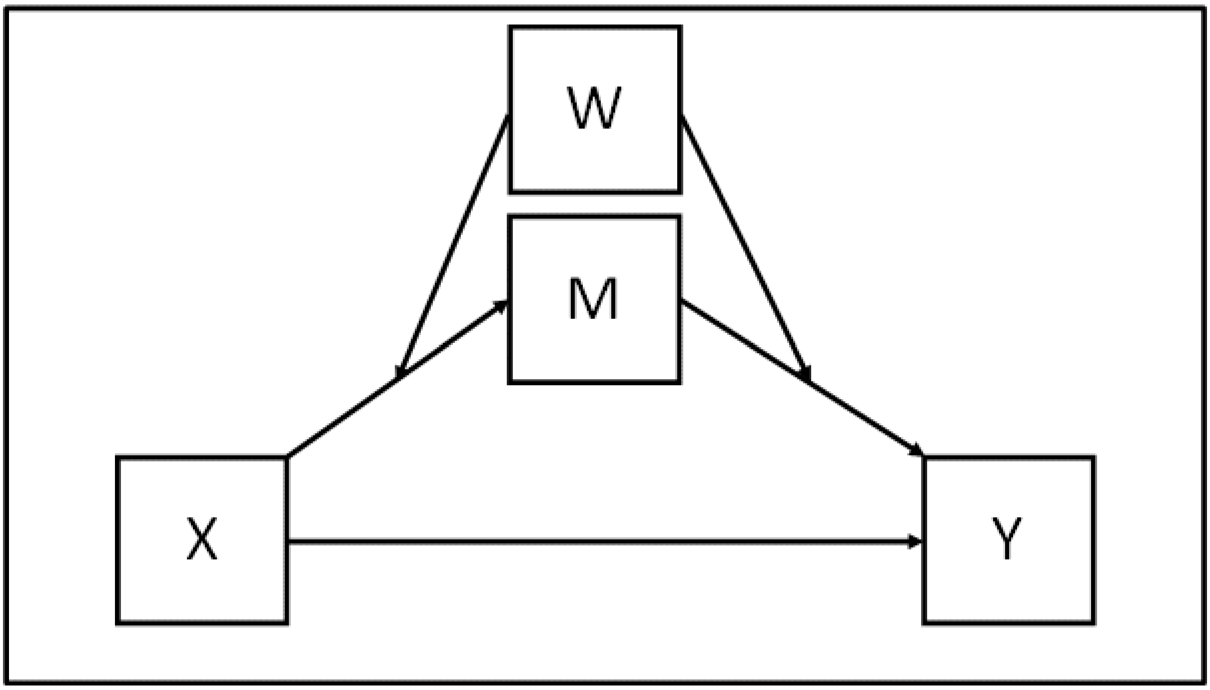
Example Variables: 1 predictor X, 1 mediator M, 1 moderator W, 1 outcome Y

Preliminary notes:

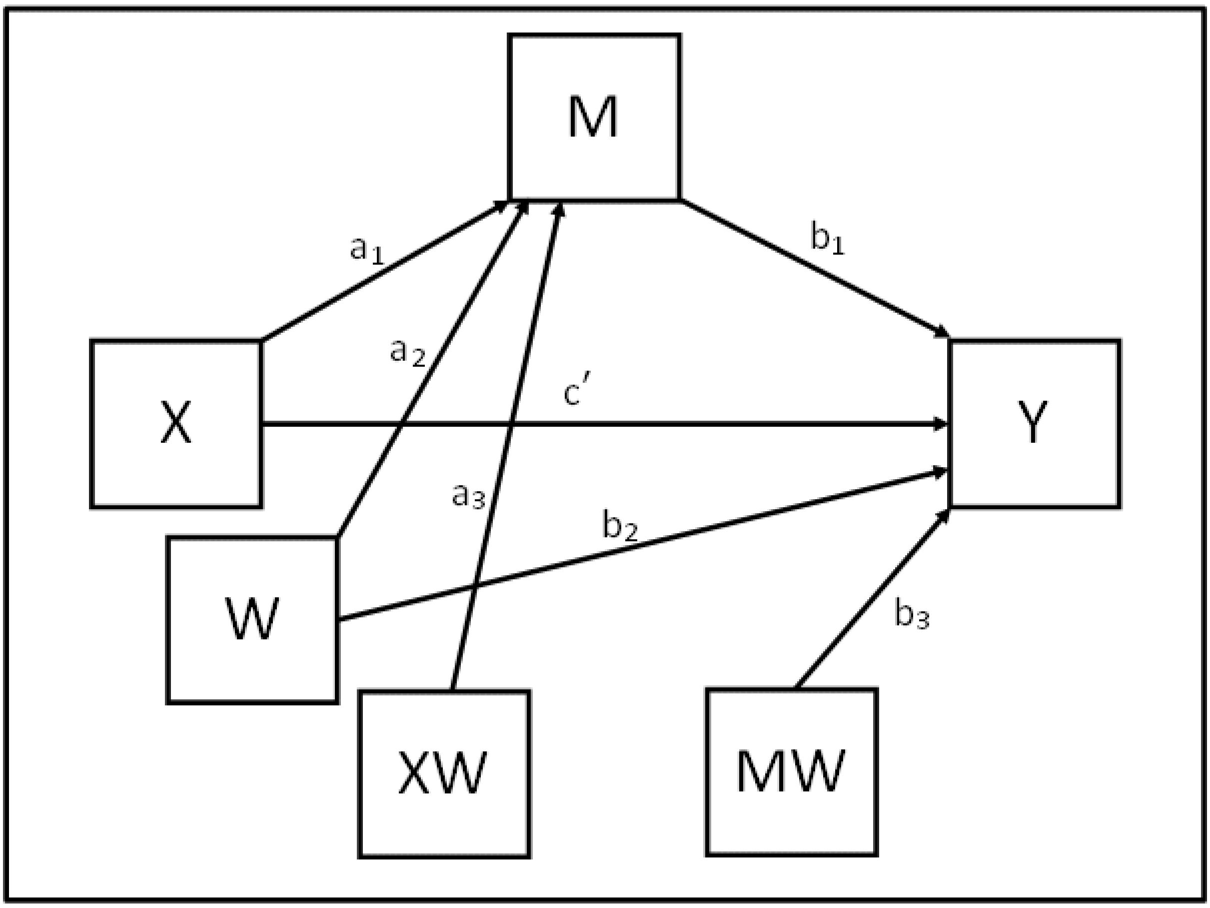
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2W + b3MW + c'X
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2W + b3MW + c'X
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2W + b3(a0 + a1X + a2W + a3XW)W + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2W + a0b3W + a1b3XW + a2b3WW + a3b3XWW + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2W + a0b3W + a2b3WW) + (a1b1 + a3b1W + a1b3W + a3b3WW + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W:

a1b1 + a3b1W + a1b3W + a3b3WW = (a1 + a3W)(b1 + b3W)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W
! Outcome variable - Y

USEVARIABLES = X M W Y XW MW;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON W (b2);
   Y ON MW (b3);

   Y ON X (cdash);

   [M] (a0);
   M ON X (a1);
```

```
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for
W
! for example, of 1 SD below mean, mean, 1 SD above mean

! 1 moderator, 3 values for it
! arbitrary naming convention for conditional indirect and
total effects used below:
! MED_Q = medium value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W
    IND_LOWW IND_MEDW IND_HIW
    TOT_LOWW TOT_MEDW TOT_HIW);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

! Calc conditional indirect effects for each combination of
moderator values

    IND_LOWW = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_W +
a3*b3*LOW_W*LOW_W;
    IND_MEDW = a1*b1 + a3*b1*MED_W + a1*b3*MED_W +
a3*b3*MED_W*MED_W;
    IND_HIW = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_W +
a3*b3*HIGH_W*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TOT_LOWW = IND_LOWW + cdash;
    TOT_MEDW = IND_MEDW + cdash;
    TOT_HIW = IND_HIW + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(LOMOD MEDMOD HIMOD);
```

```
    LOOP(XVAL,1,5,0.1);

    LOMOD = IND_LOWW*XVAL;
    MEDMOD = IND_MEDW*XVAL;
    HIMOD = IND_HIW*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 59: 1 or more mediators, in parallel if multiple (example uses 1), 1 moderators, which moderates all of the IV-Mediator path, the Mediator-DV path and the direct IV-DV path

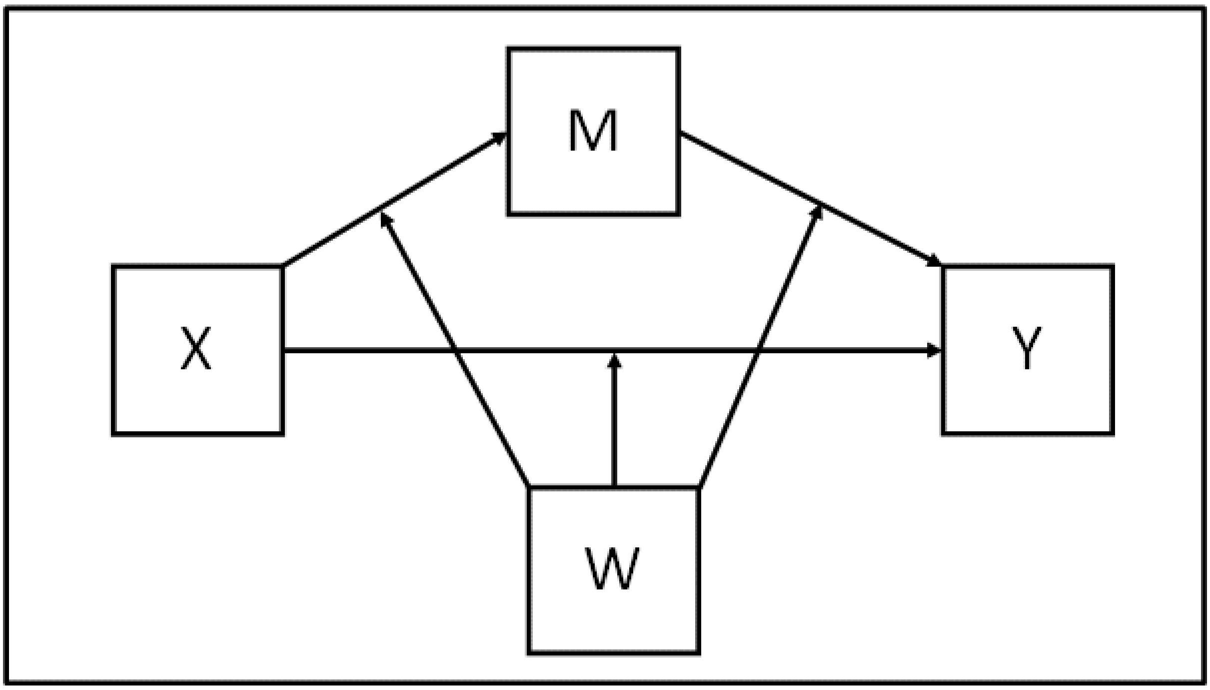
Example Variables: 1 predictor X, 1 mediator M, 1 moderator W, 1 outcome Y

Preliminary notes:

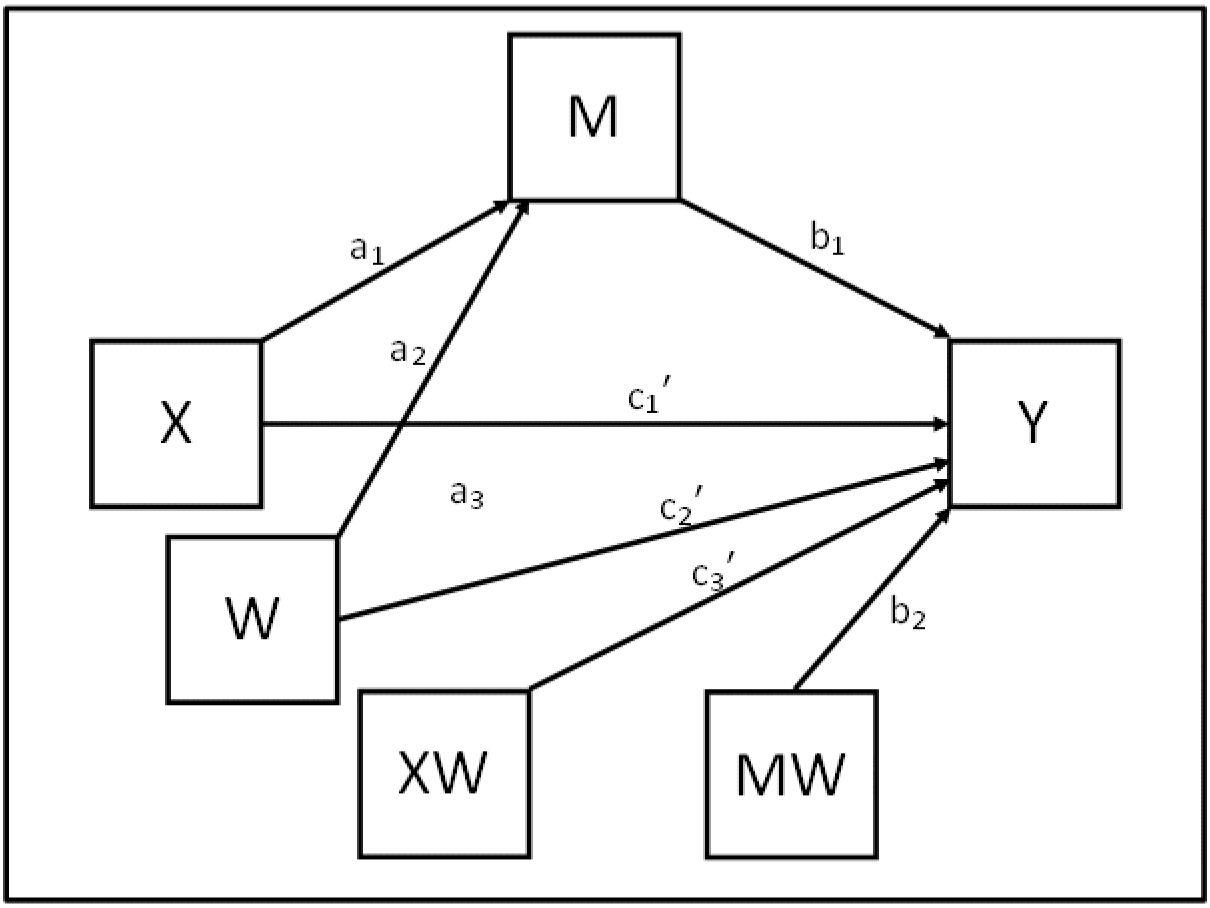
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MW + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MW + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2(a0 + a1X + a2W + a3XW)W + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + a0b2W + a1b2XW + a2b2WW + a3b2XWW + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a0b2W + a2b2WW + c2'W) + (a1b1 + a3b1W + a1b2W + a3b2WW + c1' + c3'W)X

Hence...

One indirect effect(s) of X on Y, conditional on W:

a1b1 + a3b1W + a1b2W + a3b2WW = (a1 + a3W)(b1 + b2W)

One direct effect of X on Y, conditional on W:

c1' + c3'W

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W
! Outcome variable - Y

USEVARIABLES = X M W Y XW MW;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MW (b2);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);

   [M] (a0);
   M ON X (a1);
```

```
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for
W
! for example, of 1 SD below mean, mean, 1 SD above mean

! 1 moderator, 3 values for it
! arbitrary naming convention for conditional indirect and
total effects used below:
! MED_Q = medium value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W
    IND_LOWW IND_MEDW IND_HIW
    DIR_LOWW DIR_MEDW DIR_HIW
    TOT_LOWW TOT_MEDW TOT_HIW);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

! Calc conditional indirect effects for each combination of
moderator values

    IND_LOWW = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_W +
a3*b2*LOW_W*LOW_W;
    IND_MEDW = a1*b1 + a3*b1*MED_W + a1*b2*MED_W +
a3*b2*MED_W*MED_W;
    IND_HIW = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_W +
a3*b2*HIGH_W*HIGH_W;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWW = cdash1 + cdash3*LOW_W;
    DIR_MEDW = cdash1 + cdash3*MED_W;
    DIR_HIW = cdash1 + cdash3*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TOT_LOWW = IND_LOWW + DIR_LOWW;
    TOT_MEDW = IND_MEDW + DIR_MEDW;
    TOT_HIW = IND_HIW + DIR_HIW;
```

```
! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(LOMOD MEDMOD HIMOD);

    LOOP(XVAL,1,5,0.1);

    LOMOD = IND_LOWW*XVAL;
    MEDMOD = IND_MEDW*XVAL;
    HIMOD = IND_HIW*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 60: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV- Mediator path, 1 of which also moderates the Mediator-DV path

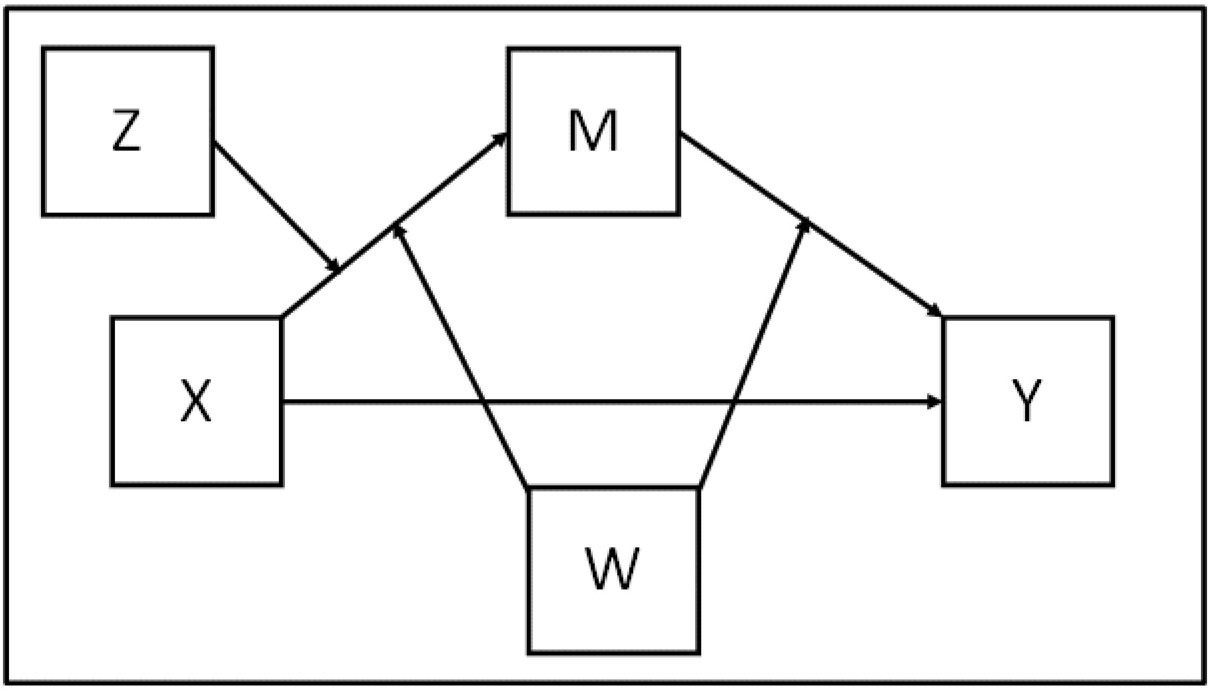
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

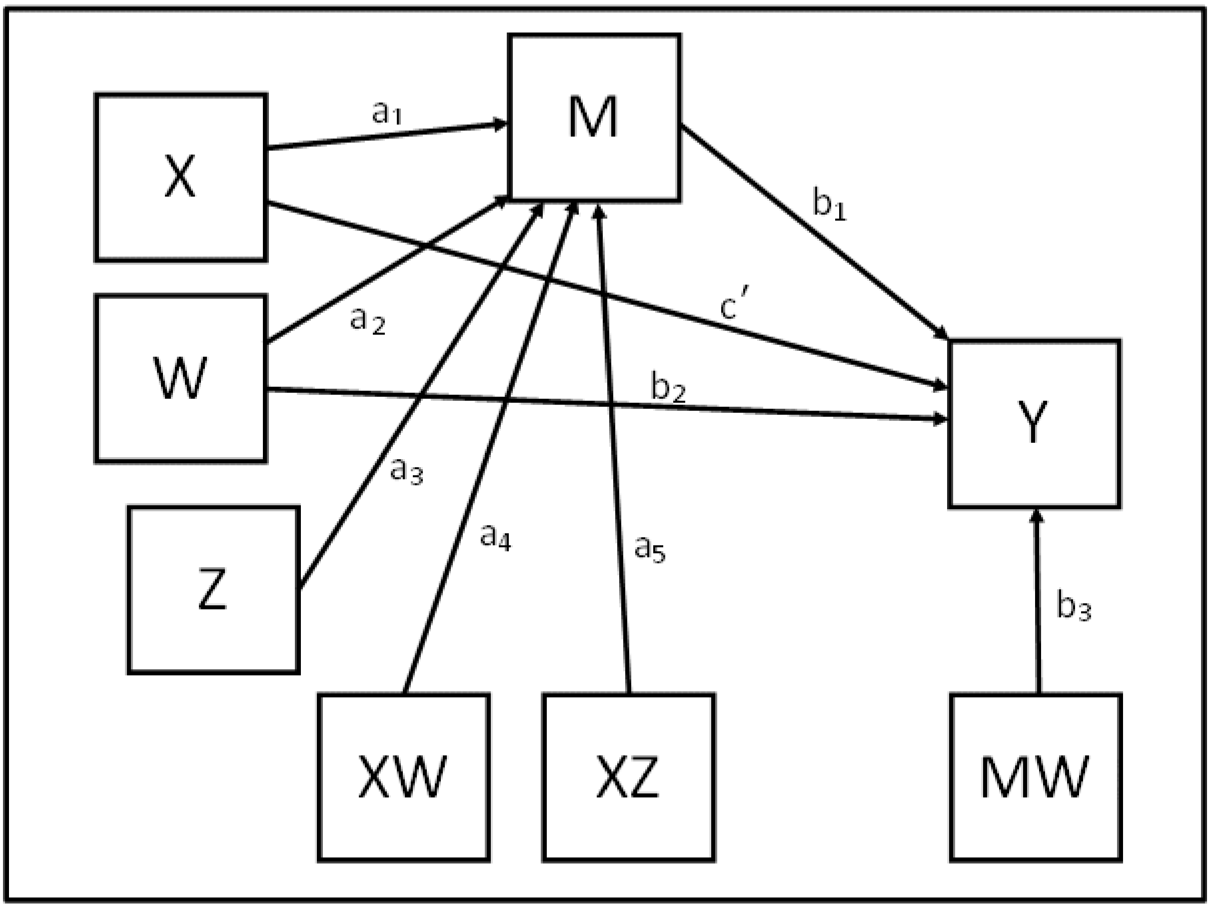
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2W + b3MW + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2W + b3MW + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2W + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ)W + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + b2W + a0b3W + a1b3XW + a2b3WW + a3b3ZW + a4b3XWW + a5b3XZW + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + b2W + a0b3W + a2b3WW + a3b3ZW) + (a1b1 + a4b1W + a5b1Z + a1b3W + a4b3WW + a5b3ZW + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a1b3W + a4b3WW + a5b3ZW = (a1 + a4W + a5Z)(b1 + b3W)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ MW;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON W (b2);
   Y ON MW (b3);

   Y ON X (cdash);
```

```
  [M] (a0);
  M ON X (a1);
  M ON W (a2);
  M ON Z (a3);
  M ON XW (a4);
  M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b3*LOW_W
+
     a4*b3*LOW_W*LOW_W + a5*b3*LOW_Z*LOW_W;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b3*MED_W
+
     a4*b3*MED_W*MED_W + a5*b3*LOW_Z*MED_W;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
```

```
a1*b3*HIGH_W +
     a4*b3*HIGH_W*HIGH_W + a5*b3*LOW_Z*HIGH_W;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b3*LOW_W
+
     a4*b3*LOW_W*LOW_W + a5*b3*MED_Z*LOW_W;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b3*MED_W
+
     a4*b3*MED_W*MED_W + a5*b3*MED_Z*MED_W;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b3*HIGH_W +
     a4*b3*HIGH_W*HIGH_W + a5*b3*MED_Z*HIGH_W;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b3*LOW_W +
     a4*b3*LOW_W*LOW_W + a5*b3*HIGH_Z*LOW_W;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b3*MED_W +
     a4*b3*MED_W*MED_W + a5*b3*HIGH_Z*MED_W;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b3*HIGH_W +
     a4*b3*HIGH_W*HIGH_W + a5*b3*HIGH_Z*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + cdash;
    TMEW_LOZ = IMEW_LOZ + cdash;
    THIW_LOZ = IHIW_LOZ + cdash;

    TLOW_MEZ = ILOW_MEZ + cdash;
    TMEW_MEZ = IMEW_MEZ + cdash;
    THIW_MEZ = IHIW_MEZ + cdash;

    TLOW_HIZ = ILOW_HIZ + cdash;
    TMEW_HIZ = IMEW_HIZ + cdash;
    THIW_HIZ = IHIW_HIZ + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);
```

```
    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 61: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV- Mediator path, 1 of which also moderates both the Mediator-DV path and the direct IV-DV path

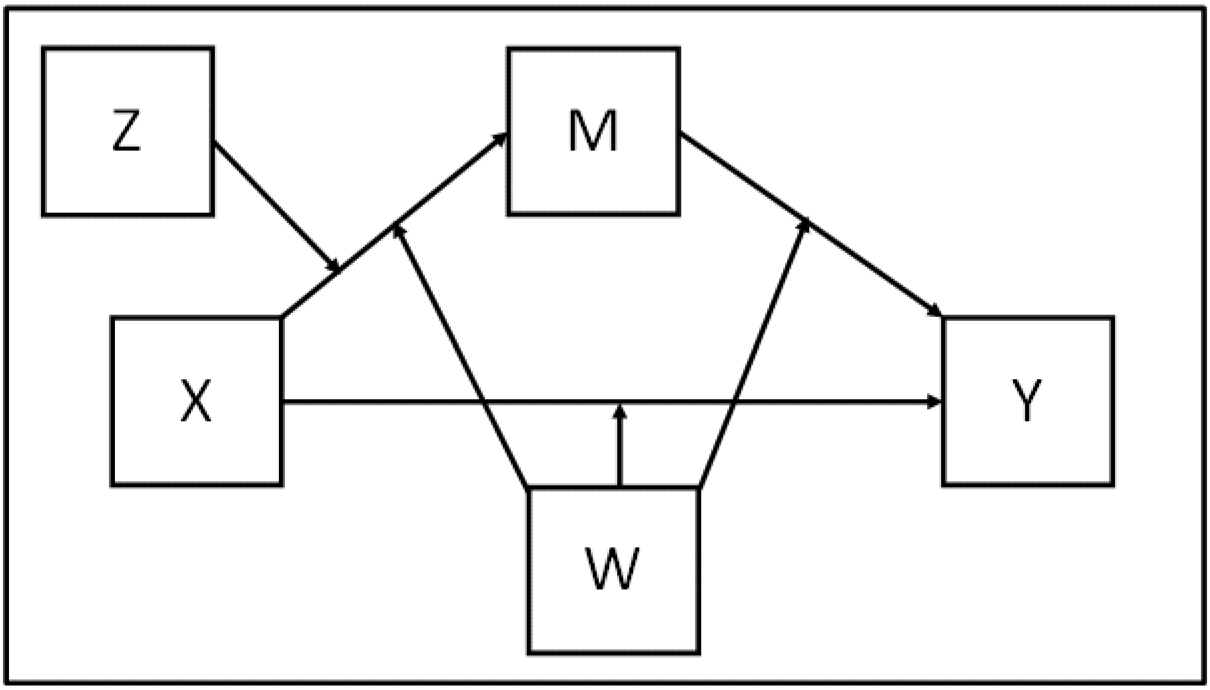
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

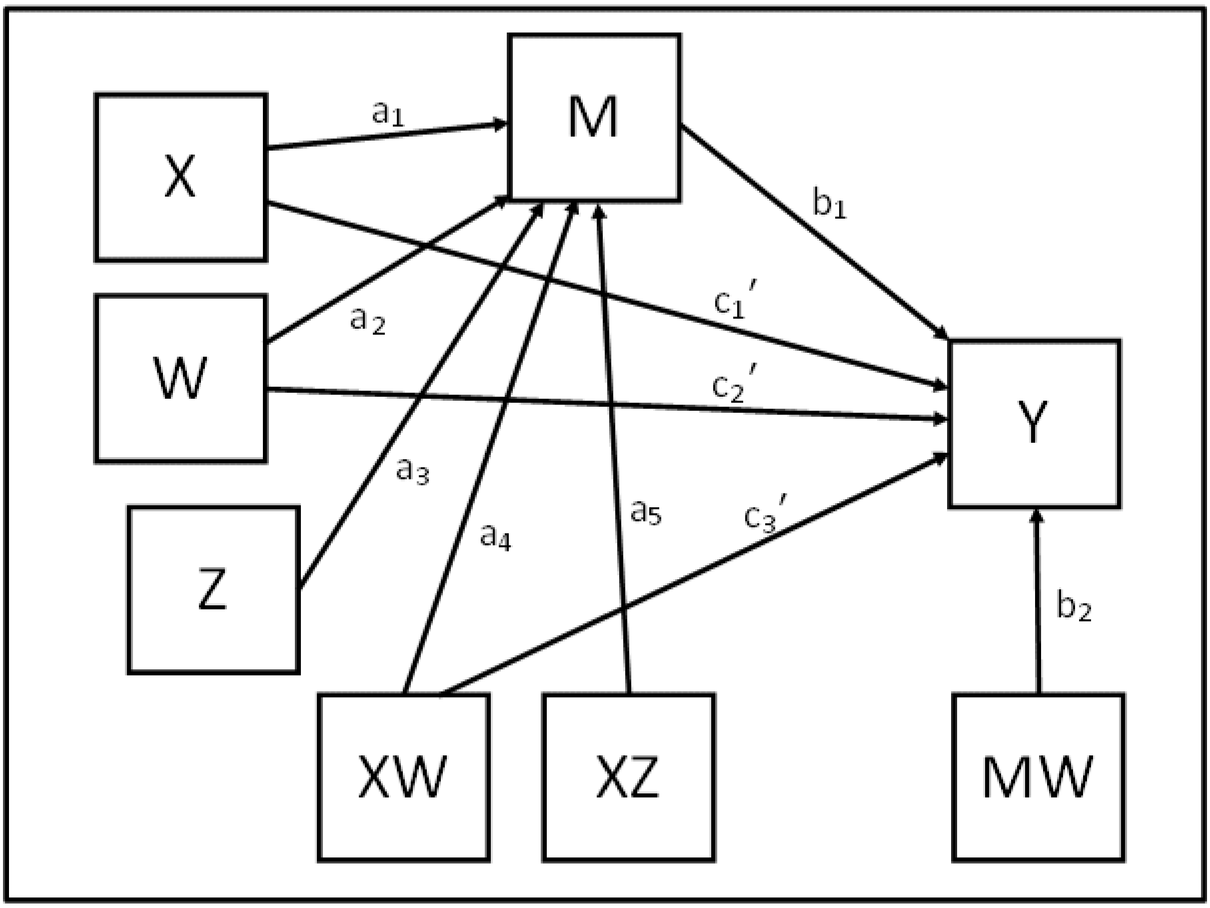
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MW + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MW + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ)W + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a0b2W + a1b2XW + a2b2WW + a3b2ZW + a4b2XWW + a5b2XZW + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a0b2W + a2b2WW + a3b2ZW + c2'W) + (a1b1 + a4b1W + a5b1Z + a1b2W + a4b2WW + a5b2ZW + c1' + c3'W)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a1b2W + a4b2WW + a5b2ZW = (a1 + a4W + a5Z)(b1 + b2W)

One direct effect of X on Y, conditional on W:

c1' + c3'W

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ MW;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MW (b2);

   Y ON X (cdash1);
   Y ON W (cdash2);
   Y ON XW (cdash3);
```

```
   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect effects
! You need to pick low, medium and high moderator values for W, Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    DIR_LOWW DIR_MEDW DIR_HIW
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your chosen high value of Z

! Calc conditional indirect effects for each combination of moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_W +
     a4*b2*LOW_W*LOW_W + a5*b2*LOW_Z*LOW_W;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_W +
     a4*b2*MED_W*MED_W + a5*b2*LOW_Z*MED_W;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
```

```
a1*b2*HIGH_W +
     a4*b2*HIGH_W*HIGH_W + a5*b2*LOW_Z*HIGH_W;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_W
+
     a4*b2*LOW_W*LOW_W + a5*b2*MED_Z*LOW_W;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_W
+
     a4*b2*MED_W*MED_W + a5*b2*MED_Z*MED_W;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b2*HIGH_W +
     a4*b2*HIGH_W*HIGH_W + a5*b2*MED_Z*HIGH_W;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b2*LOW_W +
     a4*b2*LOW_W*LOW_W + a5*b2*HIGH_Z*LOW_W;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b2*MED_W +
     a4*b2*MED_W*MED_W + a5*b2*HIGH_Z*MED_W;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b2*HIGH_W +
     a4*b2*HIGH_W*HIGH_W + a5*b2*HIGH_Z*HIGH_W;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWW = cdash1 + cdash3*LOW_W;
    DIR_MEDW = cdash1 + cdash3*MED_W;
    DIR_HIW = cdash1 + cdash3*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + DIR_LOWW;
    TMEW_LOZ = IMEW_LOZ + DIR_MEDW;
    THIW_LOZ = IHIW_LOZ + DIR_HIW;

    TLOW_MEZ = ILOW_MEZ + DIR_LOWW;
    TMEW_MEZ = IMEW_MEZ + DIR_MEDW;
    THIW_MEZ = IHIW_MEZ + DIR_HIW;

    TLOW_HIZ = ILOW_HIZ + DIR_LOWW;
    TMEW_HIZ = IMEW_HIZ + DIR_MEDW;
    THIW_HIZ = IHIW_HIZ + DIR_HIW;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
```

```
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 62: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV- Mediator path, 1 of which also moderates the Mediator-DV path, the with the other moderating the direct IV-DV path

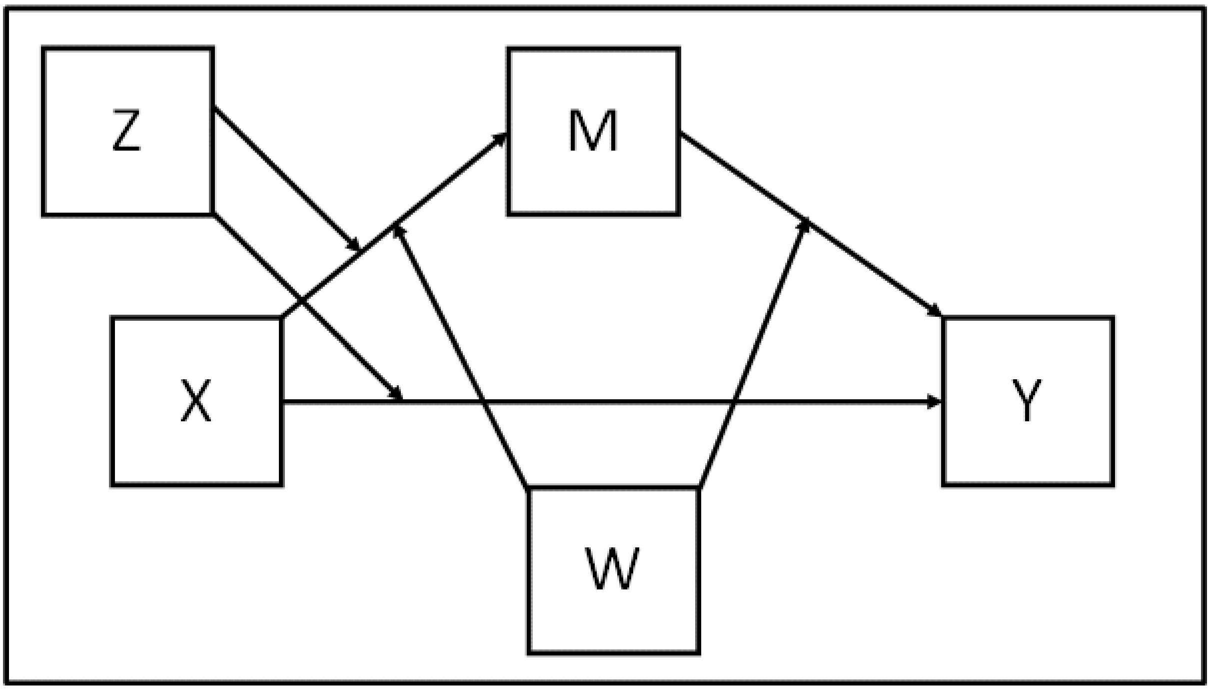
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

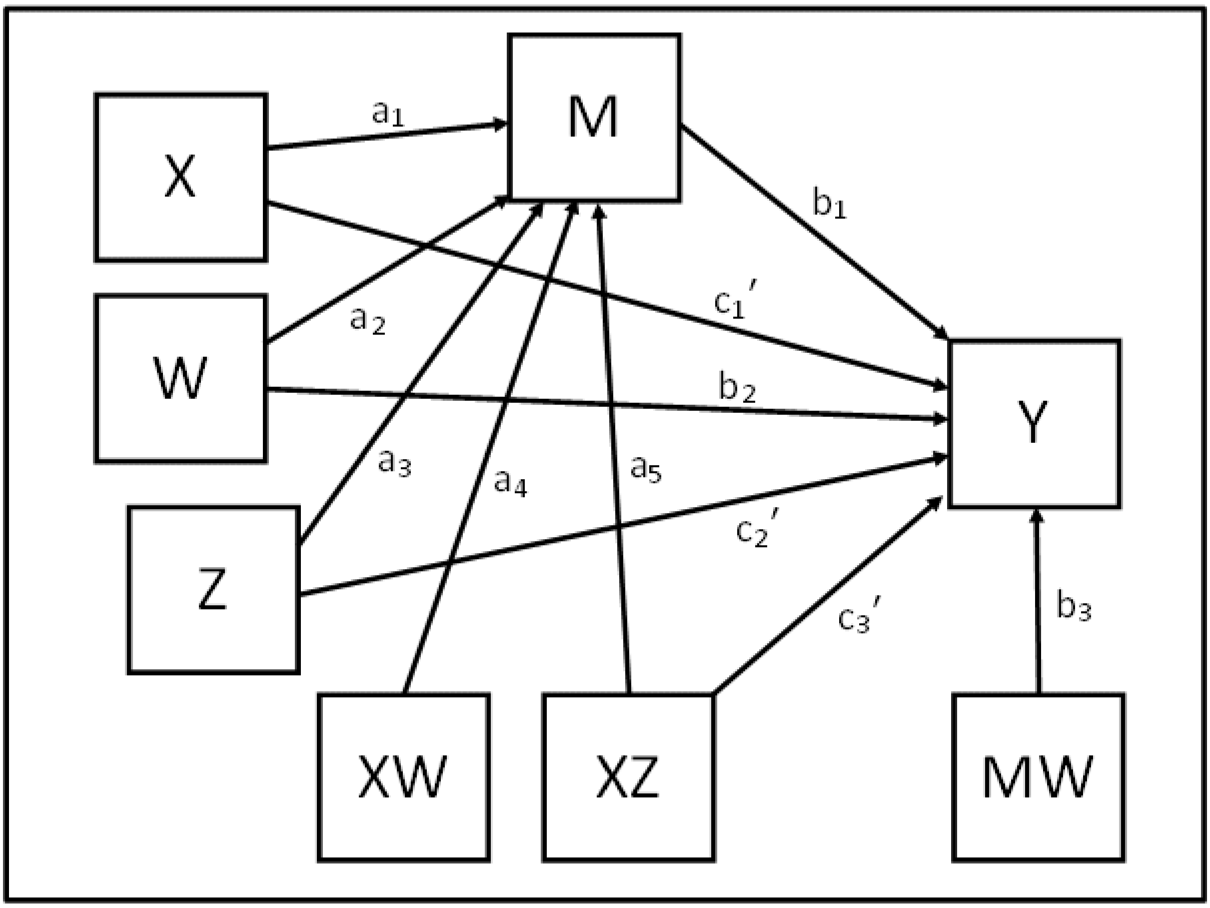
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2W + b3MW + c1'X + c2'Z + c3'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2W + b3MW + c1'X + c2'Z + c3'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2W + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ)W + c1'X + c2'Z + c3'XZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + b2W + a0b3W + a1b3XW + a2b3WW + a3b3ZW + a4b3XWW + a5b3XZW + c1'X + c2'Z + c3'XZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + b2W + a0b3W + a2b3WW + a3b3ZW + c2'Z) + (a1b1 + a4b1W + a5b1Z + a1b3W + a4b3WW + a5b3ZW + c1' + c3'Z)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a1b3W + a4b3WW + a5b3ZW = (a1 + a4W + a5Z)(b1 + b3W)

One direct effect of X on Y, conditional on Z:

c1' + c3'Z

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ MW;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON W (b2);
   Y ON MW (b3);
```

```
    Y ON X (cdash1);
    Y ON Z (cdash2);
    Y ON XZ (cdash3);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    DIR_LOWZ DIR_MEDZ DIR_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b3*LOW_W
+
     a4*b3*LOW_W*LOW_W + a5*b3*LOW_Z*LOW_W;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b3*MED_W
```

```
+
     a4*b3*MED_W*MED_W + a5*b3*LOW_Z*MED_W;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a1*b3*HIGH_W +
     a4*b3*HIGH_W*HIGH_W + a5*b3*LOW_Z*HIGH_W;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b3*LOW_W
+
     a4*b3*LOW_W*LOW_W + a5*b3*MED_Z*LOW_W;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b3*MED_W
+
     a4*b3*MED_W*MED_W + a5*b3*MED_Z*MED_W;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b3*HIGH_W +
     a4*b3*HIGH_W*HIGH_W + a5*b3*MED_Z*HIGH_W;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b3*LOW_W +
     a4*b3*LOW_W*LOW_W + a5*b3*HIGH_Z*LOW_W;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b3*MED_W +
     a4*b3*MED_W*MED_W + a5*b3*HIGH_Z*MED_W;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b3*HIGH_W +
     a4*b3*HIGH_W*HIGH_W + a5*b3*HIGH_Z*HIGH_W;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWZ = cdash1 + cdash3*LOW_Z;
    DIR_MEDZ = cdash1 + cdash3*MED_Z;
    DIR_HIZ = cdash1 + cdash3*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + DIR_LOWZ;
    TMEW_LOZ = IMEW_LOZ + DIR_LOWZ;
    THIW_LOZ = IHIW_LOZ + DIR_LOWZ;

    TLOW_MEZ = ILOW_MEZ + DIR_MEDZ;
    TMEW_MEZ = IMEW_MEZ + DIR_MEDZ;
    THIW_MEZ = IHIW_MEZ + DIR_MEDZ;

    TLOW_HIZ = ILOW_HIZ + DIR_HIZ;
    TMEW_HIZ = IMEW_HIZ + DIR_HIZ;
    THIW_HIZ = IHIW_HIZ + DIR_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
```

```
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 63: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV- Mediator path and the direct IV-DV path, 1 of which also moderates the Mediator-DV path

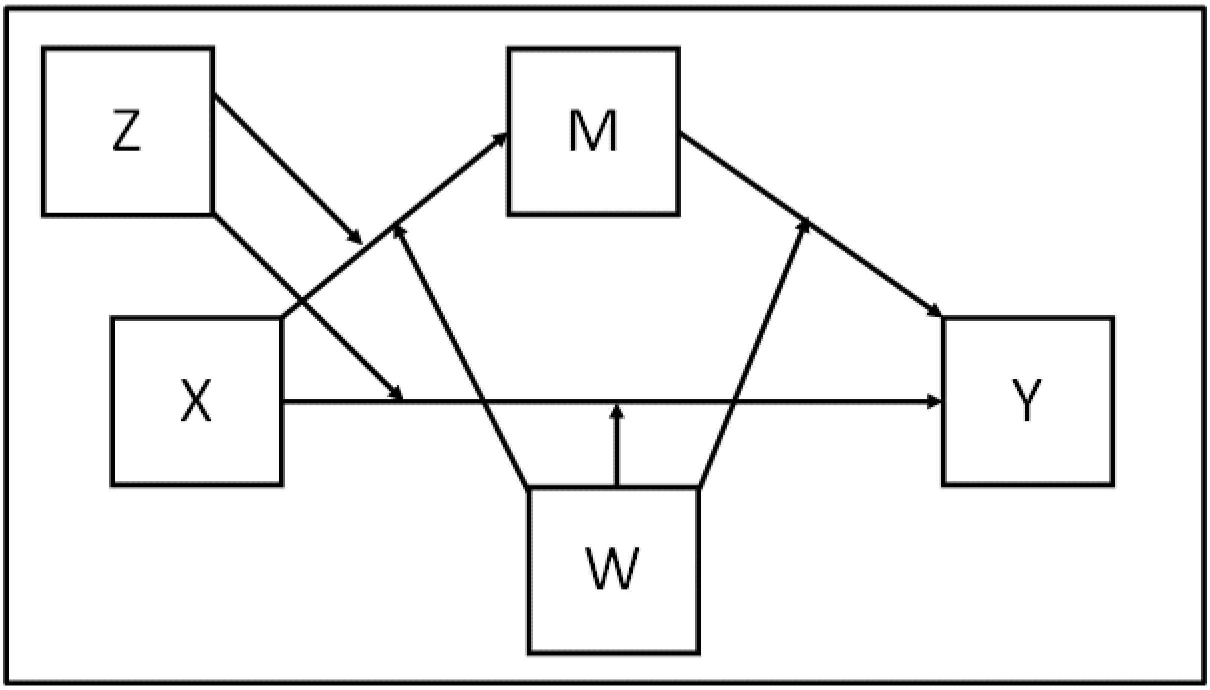
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

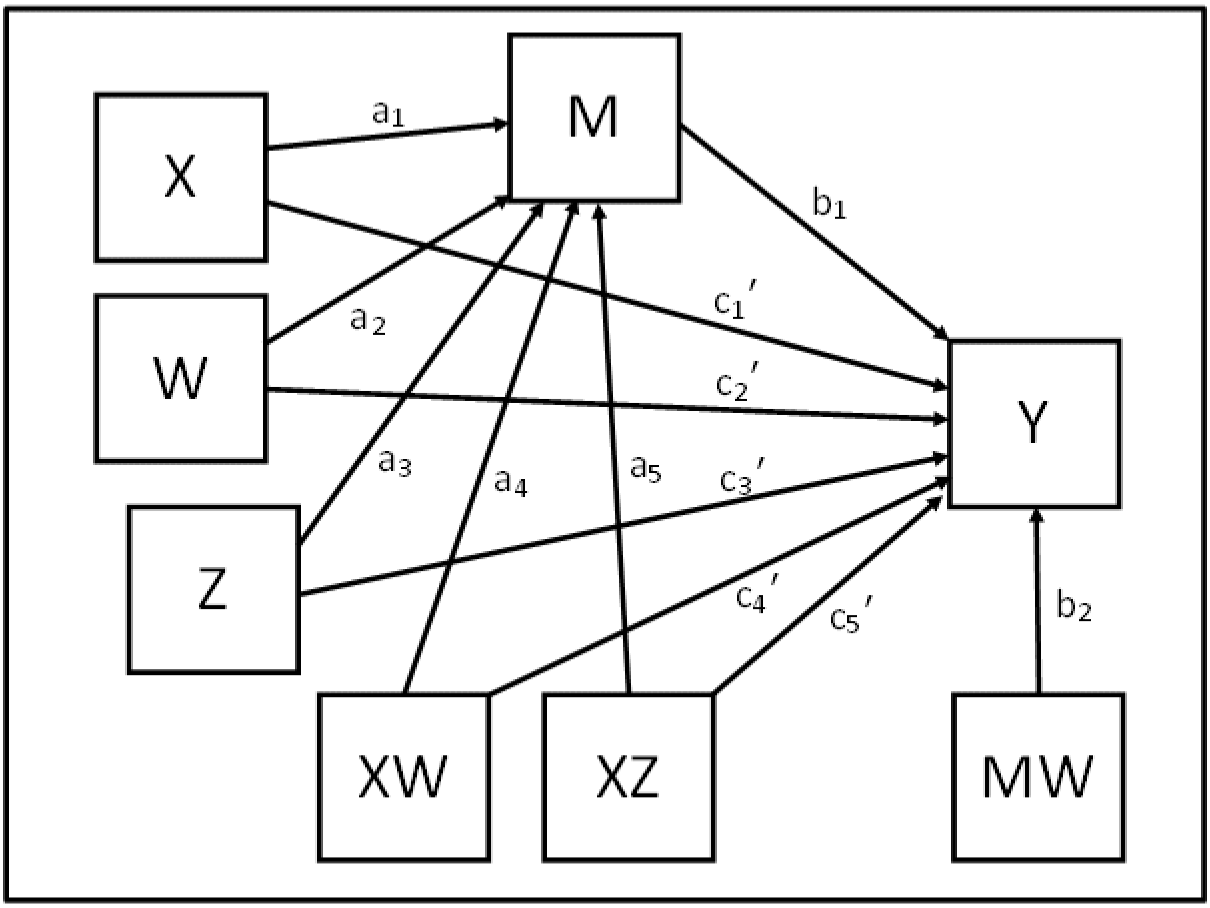
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MW + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MW + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ)W + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a0b2W + a1b2XW + a2b2WW + a3b2ZW + a4b2XWW + a5b2XZW + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a0b2W + a2b2WW + a3b2ZW + c2'W + c3'Z) + (a1b1 + a4b1W + a5b1Z + a1b2W + a4b2WW + a5b2ZW + c1' + c4'W + c5'Z)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a1b2W + a4b2WW + a5b2ZW = (a1 + a4W + a5Z)(b1 + b2W)

One direct effect of X on Y, conditional on W, Z:

c1' + c4'W + c5'Z

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ MW;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MW (b2);
```

```
    Y ON X (cdash1);
    Y ON W (cdash2);
    Y ON Z (cdash3);
    Y ON XW (cdash4);
    Y ON XZ (cdash5);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
     NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
     ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
     ILOW_HIZ IMEW_HIZ IHIW_HIZ
     DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
     DLOW_HIZ DMEW_HIZ DHIW_HIZ
     TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
     TLOW_HIZ TMEW_HIZ THIW_HIZ);

     LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
     MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
     HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

     LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
     MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
     HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values
```

```
    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_W
+
     a4*b2*LOW_W*LOW_W + a5*b2*LOW_Z*LOW_W;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_W
+
     a4*b2*MED_W*MED_W + a5*b2*LOW_Z*MED_W;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a1*b2*HIGH_W +
     a4*b2*HIGH_W*HIGH_W + a5*b2*LOW_Z*HIGH_W;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_W
+
     a4*b2*LOW_W*LOW_W + a5*b2*MED_Z*LOW_W;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_W
+
     a4*b2*MED_W*MED_W + a5*b2*MED_Z*MED_W;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b2*HIGH_W +
     a4*b2*HIGH_W*HIGH_W + a5*b2*MED_Z*HIGH_W;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b2*LOW_W +
     a4*b2*LOW_W*LOW_W + a5*b2*HIGH_Z*LOW_W;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b2*MED_W +
     a4*b2*MED_W*MED_W + a5*b2*HIGH_Z*MED_W;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b2*HIGH_W +
     a4*b2*HIGH_W*HIGH_W + a5*b2*HIGH_Z*HIGH_W;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + DLOW_LOZ;
    TMEW_LOZ = IMEW_LOZ + DMEW_LOZ;
    THIW_LOZ = IHIW_LOZ + DHIW_LOZ;
```

```
    TLOW_MEZ = ILOW_MEZ + DLOW_MEZ;
    TMEW_MEZ = IMEW_MEZ + DMEW_MEZ;
    THIW_MEZ = IHIW_MEZ + DHIW_MEZ;

    TLOW_HIZ = ILOW_HIZ + DLOW_HIZ;
    TMEW_HIZ = IMEW_HIZ + DMEW_HIZ;
    THIW_HIZ = IHIW_HIZ + DHIW_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 64: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the Mediator-DV path, 1 of which also moderates the IV-Mediator path

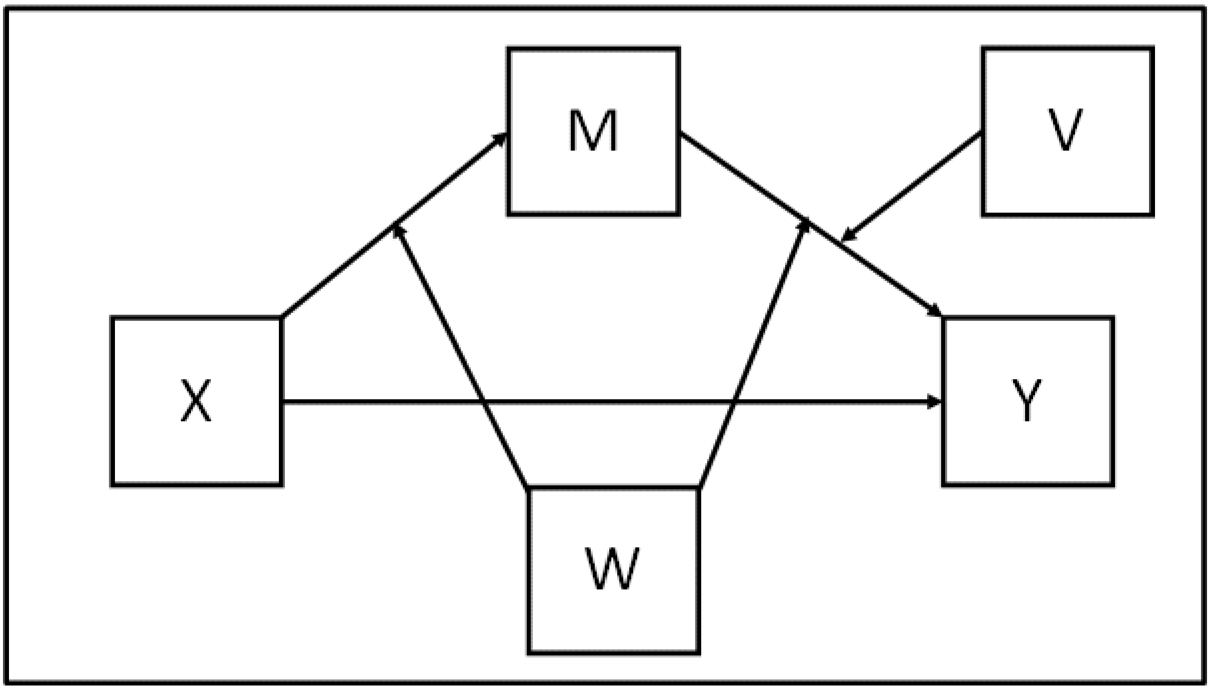
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

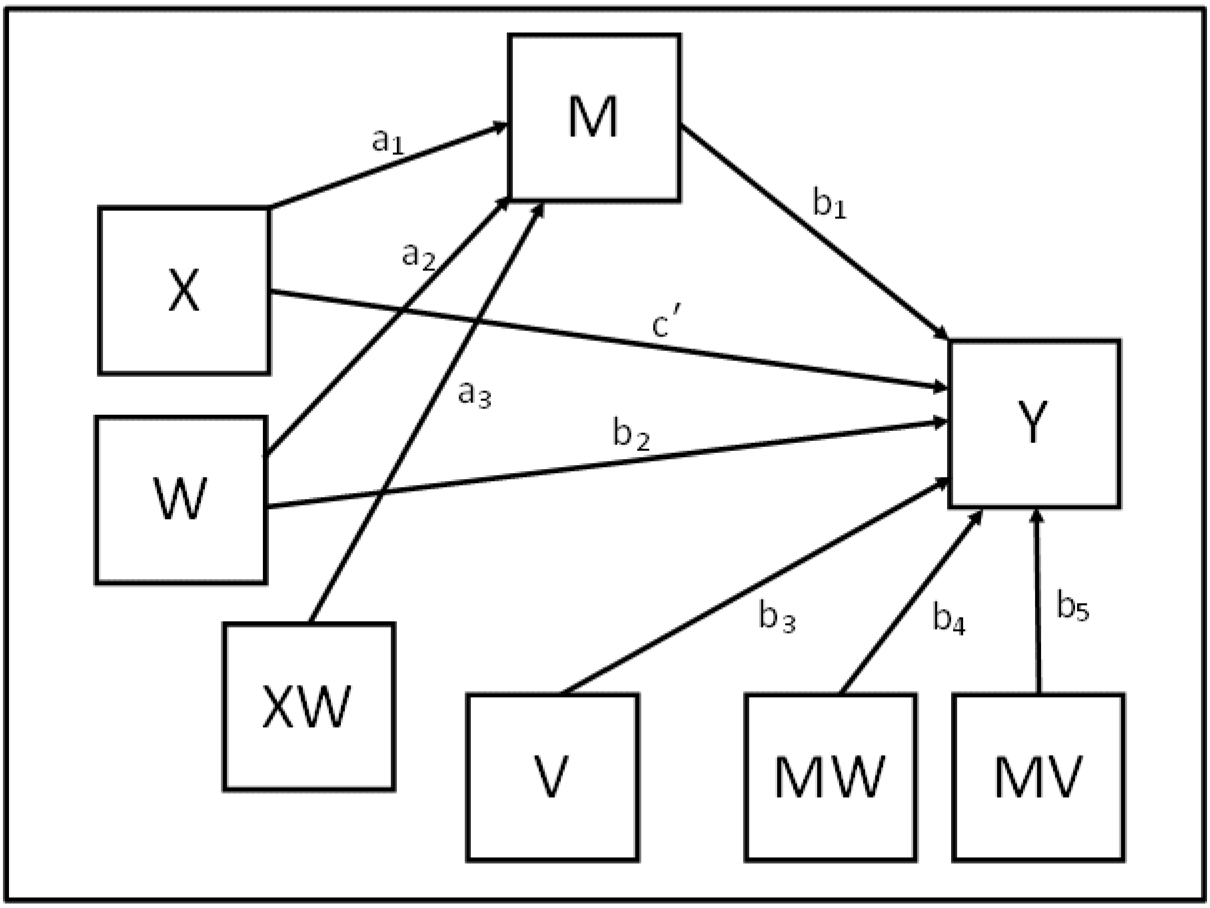
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2W + b3V + b4MW + b5MV + c'X
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2W + b3V + b4MW + b5MV + c'X
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2W + b3V + b4(a0 + a1X + a2W + a3XW)W + b5(a0 + a1X + a2W + a3XW)V + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2W + b3V + a0b4W + a1b4XW + a2b4WW + a3b4XWW + a0b5V + a1b5XV + a2b5WV + a3b5XWV + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2W + b3V + a0b4W + a2b4WW + a0b5V + a2b5WV) + (a1b1 + a3b1W + a1b4W + a3b4WW + a1b5V + a3b5WV + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b4W + a3b4WW + a1b5V + a3b5WV = (a1 + a3W)(b1 + b4W + b5V)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) - W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW MW MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MV = M*V;
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON W (b2);
   Y ON V (b3);
   Y ON MW (b4);
   Y ON MV (b5);
```

```
    Y ON X (cdash);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
    ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
    ILOW_HIV IMEW_HIV IHIW_HIV
    TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
    TLOW_HIV TMEW_HIV THIW_HIV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_W +
a3*b4*LOW_W*LOW_W +
     a1*b5*LOW_V + a3*b5*LOW_W*LOW_V;
    IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b4*MED_W +
a3*b4*MED_W*MED_W +
     a1*b5*LOW_V + a3*b5*MED_W*LOW_V;
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_W +
a3*b4*HIGH_W*HIGH_W +
     a1*b5*LOW_V + a3*b5*HIGH_W*LOW_V;
```

```
    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_W +
a3*b4*LOW_W*LOW_W +
     a1*b5*MED_V + a3*b5*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b4*MED_W +
a3*b4*MED_W*MED_W +
     a1*b5*MED_V + a3*b5*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_W +
a3*b4*HIGH_W*HIGH_W +
     a1*b5*MED_V + a3*b5*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_W +
a3*b4*LOW_W*LOW_W +
     a1*b5*HIGH_V + a3*b5*LOW_W*HIGH_V;
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b4*MED_W +
a3*b4*MED_W*MED_W +
     a1*b5*HIGH_V + a3*b5*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_W +
a3*b4*HIGH_W*HIGH_W +
     a1*b5*HIGH_V + a3*b5*HIGH_W*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + cdash;
    TMEW_LOV = IMEW_LOV + cdash;
    THIW_LOV = IHIW_LOV + cdash;

    TLOW_MEV = ILOW_MEV + cdash;
    TMEW_MEV = IMEW_MEV + cdash;
    THIW_MEV = IHIW_MEV + cdash;

    TLOW_HIV = ILOW_HIV + cdash;
    TMEW_HIV = IMEW_HIV + cdash;
    THIW_HIV = IHIW_HIV + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);
```

```
    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;

    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 65: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the Mediator-DV path, 1 of which also moderates both the IV-Mediator path and the direct IV-DV path

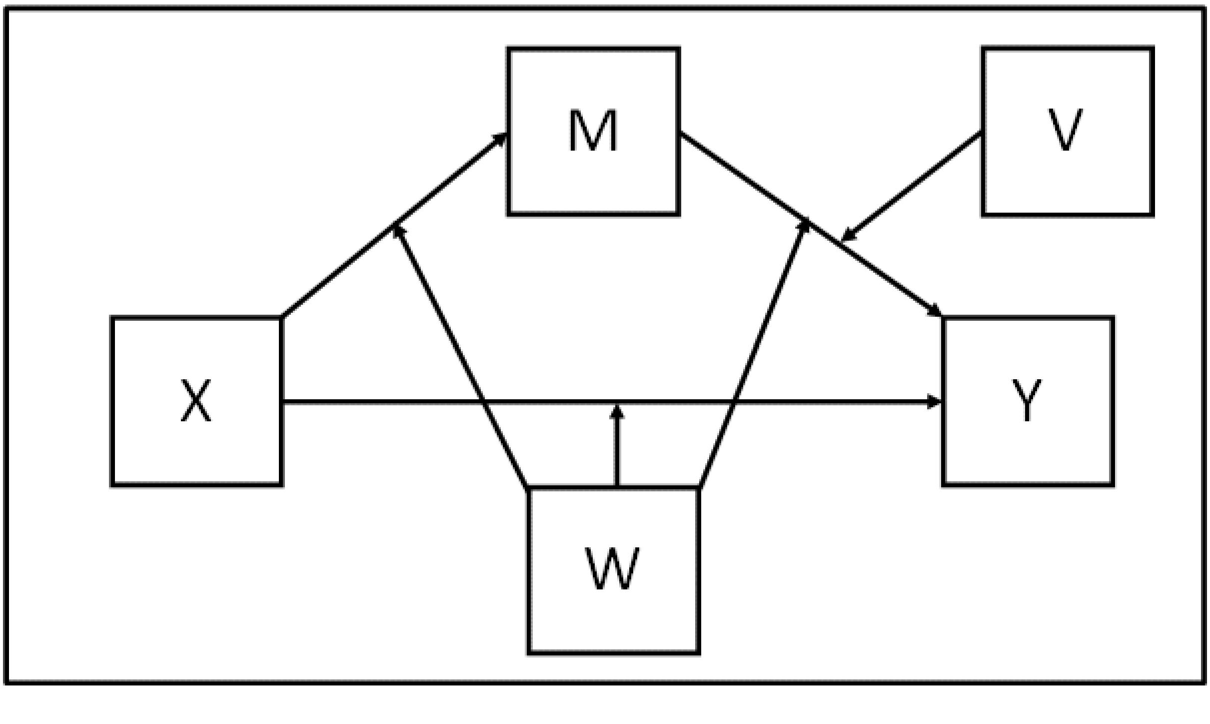
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

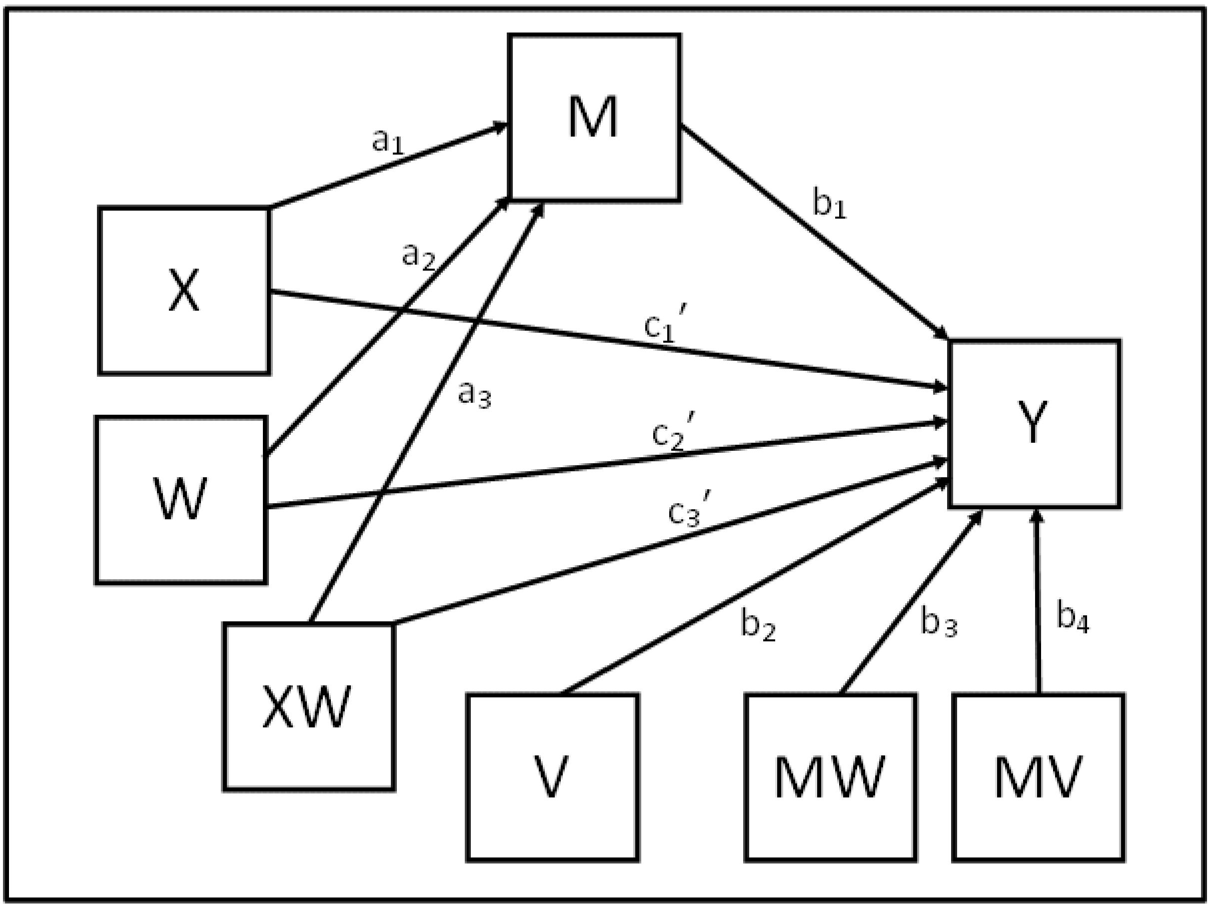
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2V + b3MW + b4MV + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2V + b3MW + b4MV + c1'X + c2'W + c3'XW
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2V + b3(a0 + a1X + a2W + a3XW)W + b4(a0 + a1X + a2W + a3XW)V + c1'X + c2'W + c3'XW

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2V + a0b3W + a1b3XW + a2b3WW + a3b3XWW + a0b4V + a1b4XV + a2b4WV + a3b4XWV + c1'X + c2'W + c3'XW

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2V + a0b3W + a2b3WW + a0b4V + a2b4W + c2'W) + (a1b1 + a3b1W + a1b3W + a3b3WW + a1b4V + a3b4WV + c1' + c3'W)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b3W + a3b3WW + a1b4V + a3b4WV = (a1 + a3W)(b1 + b3W + b4V)

One direct effect of X on Y, conditional on W:

c1' + c3'W

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW MW MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MV = M*V;
   XW = X*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON V (b2);
   Y ON MW (b3);
   Y ON MV (b4);
```

```
    Y ON X (cdash1);
    Y ON W (cdash2);
    Y ON XW (cdash3);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
     NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
     ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
     ILOW_HIV IMEW_HIV IHIW_HIV
     DIR_LOWW DIR_MEDW DIR_HIW
     TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
     TLOW_HIV TMEW_HIV THIW_HIV);

     LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
     MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
     HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

     LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
     MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
     HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

     ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_W +
a3*b3*LOW_W*LOW_W +
      a1*b4*LOW_V + a3*b4*LOW_W*LOW_V;
     IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b3*MED_W +
a3*b3*MED_W*MED_W +
      a1*b4*LOW_V + a3*b4*MED_W*LOW_V;
```

```
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_W +
a3*b3*HIGH_W*HIGH_W +
     a1*b4*LOW_V + a3*b4*HIGH_W*LOW_V;

    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_W +
a3*b3*LOW_W*LOW_W +
     a1*b4*MED_V + a3*b4*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b3*MED_W +
a3*b3*MED_W*MED_W +
     a1*b4*MED_V + a3*b4*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_W +
a3*b3*HIGH_W*HIGH_W +
     a1*b4*MED_V + a3*b4*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_W +
a3*b3*LOW_W*LOW_W +
     a1*b4*HIGH_V + a3*b4*LOW_W*HIGH_V;
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b3*MED_W +
a3*b3*MED_W*MED_W +
     a1*b4*HIGH_V + a3*b4*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_W +
a3*b3*HIGH_W*HIGH_W +
     a1*b4*HIGH_V + a3*b4*HIGH_W*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWW = cdash1 + cdash3*LOW_W;
    DIR_MEDW = cdash1 + cdash3*MED_W;
    DIR_HIW = cdash1 + cdash3*HIGH_W;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + DIR_LOWW;
    TMEW_LOV = IMEW_LOV + DIR_MEDW;
    THIW_LOV = IHIW_LOV + DIR_HIW;

    TLOW_MEV = ILOW_MEV + DIR_LOWW;
    TMEW_MEV = IMEW_MEV + DIR_MEDW;
    THIW_MEV = IHIW_MEV + DIR_HIW;

    TLOW_HIV = ILOW_HIV + DIR_LOWW;
    TMEW_HIV = IMEW_HIV + DIR_MEDW;
    THIW_HIV = IHIW_HIV + DIR_HIW;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
```

```
by
! logical min and max limits of predictor X used in analysis
    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;

    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 66: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the Mediator-DV path, 1 of which also moderates the IV-Mediator path, the with the other moderating the direct IV-DV path

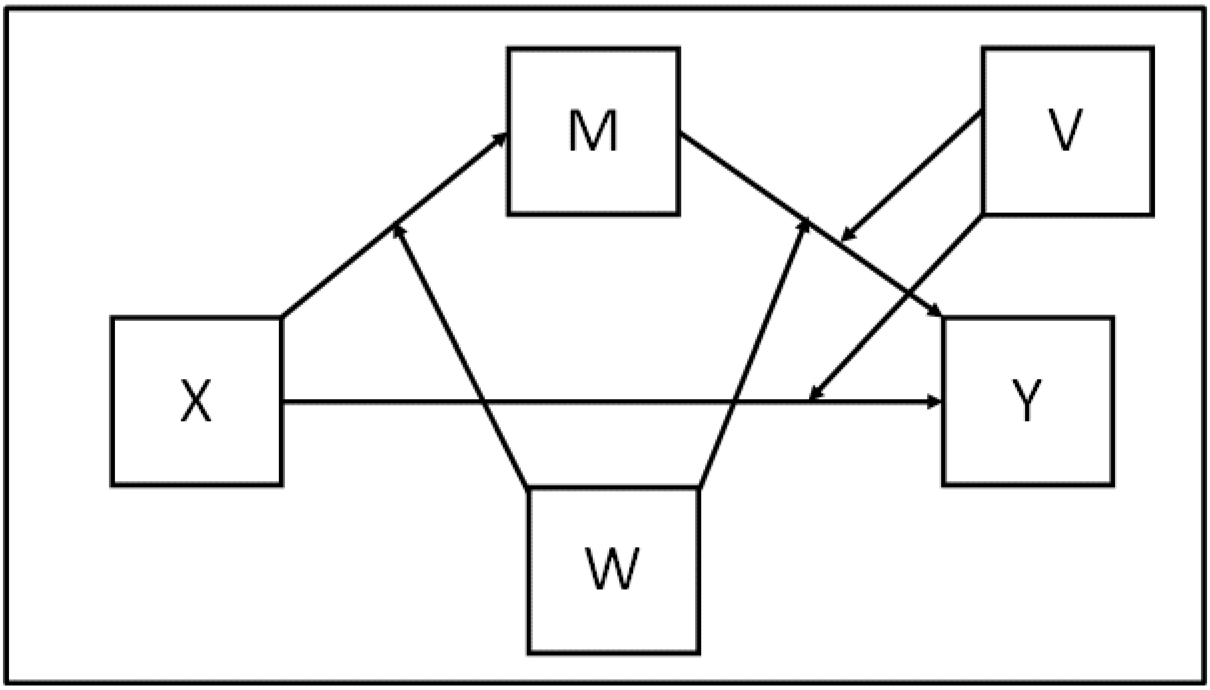
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

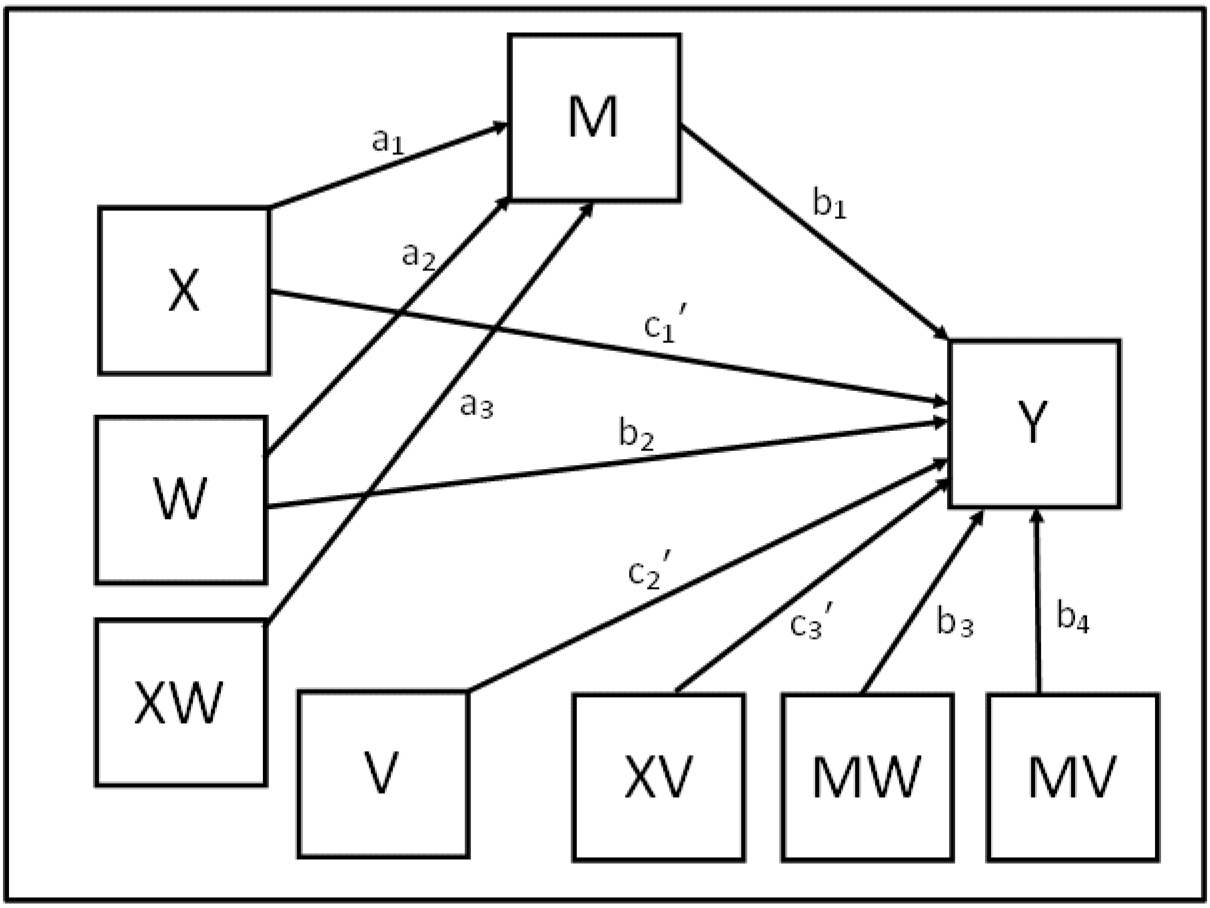
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2W + b3MW + b4MV + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2W + b3MW + b4MV + c1'X + c2'V + c3'XV
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2W + b3(a0 + a1X + a2W + a3XW)W + b4(a0 + a1X + a2W + a3XW)V + c1'X + c2'V + c3'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2W + a0b3W + a1b3XW + a2b3WW + a3b3XWW + a0b4V + a1b4XV + a2b4WV + a3b4XWV + c1'X + c2'V + c3'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2W + a0b3W + a2b3WW + a0b4V + a2b4WV + c2'V) + (a1b1 + a3b1W + a1b3W + a3b3WW + a1b4V + a3b4WV + c1' + c3'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b3W + a3b3WW + a1b4V + a3b4WV = (a1 + a3W)(b1 + b3W + b4V)

One direct effect of X on Y, conditional on V:

c1' + c3'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW XV MW MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MV = M*V;
   XW = X*W;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON W (b2);
   Y ON MW (b3);
   Y ON MV (b4);
```

```
    Y ON X (cdash1);
    Y ON V (cdash2);
    Y ON XV (cdash3);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
    ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
    ILOW_HIV IMEW_HIV IHIW_HIV
    DIR_LOWV DIR_MEDV DIR_HIV
    TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
    TLOW_HIV TMEW_HIV THIW_HIV);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_W +
a3*b3*LOW_W*LOW_W +
     a1*b4*LOW_V + a3*b4*LOW_W*LOW_V;
    IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b3*MED_W +
a3*b3*MED_W*MED_W +
     a1*b4*LOW_V + a3*b4*MED_W*LOW_V;
```

```
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_W +
a3*b3*HIGH_W*HIGH_W +
     a1*b4*LOW_V + a3*b4*HIGH_W*LOW_V;

    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_W +
a3*b3*LOW_W*LOW_W +
     a1*b4*MED_V + a3*b4*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b3*MED_W +
a3*b3*MED_W*MED_W +
     a1*b4*MED_V + a3*b4*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_W +
a3*b3*HIGH_W*HIGH_W +
     a1*b4*MED_V + a3*b4*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b3*LOW_W +
a3*b3*LOW_W*LOW_W +
     a1*b4*HIGH_V + a3*b4*LOW_W*HIGH_V;
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b3*MED_W +
a3*b3*MED_W*MED_W +
     a1*b4*HIGH_V + a3*b4*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b3*HIGH_W +
a3*b3*HIGH_W*HIGH_W +
     a1*b4*HIGH_V + a3*b4*HIGH_W*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DIR_LOWV = cdash1 + cdash3*LOW_V;
    DIR_MEDV = cdash1 + cdash3*MED_V;
    DIR_HIV = cdash1 + cdash3*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + DIR_LOWV;
    TMEW_LOV = IMEW_LOV + DIR_LOWV;
    THIW_LOV = IHIW_LOV + DIR_LOWV;

    TLOW_MEV = ILOW_MEV + DIR_MEDV;
    TMEW_MEV = IMEW_MEV + DIR_MEDV;
    THIW_MEV = IHIW_MEV + DIR_MEDV;

    TLOW_HIV = ILOW_HIV + DIR_HIV;
    TMEW_HIV = IMEW_HIV + DIR_HIV;
    THIW_HIV = IHIW_HIV + DIR_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
```

```
by
! logical min and max limits of predictor X used in analysis
    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;

    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 67: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the Mediator-DV path and the direct IV-DV path, 1 of which also moderates the IV-Mediator path

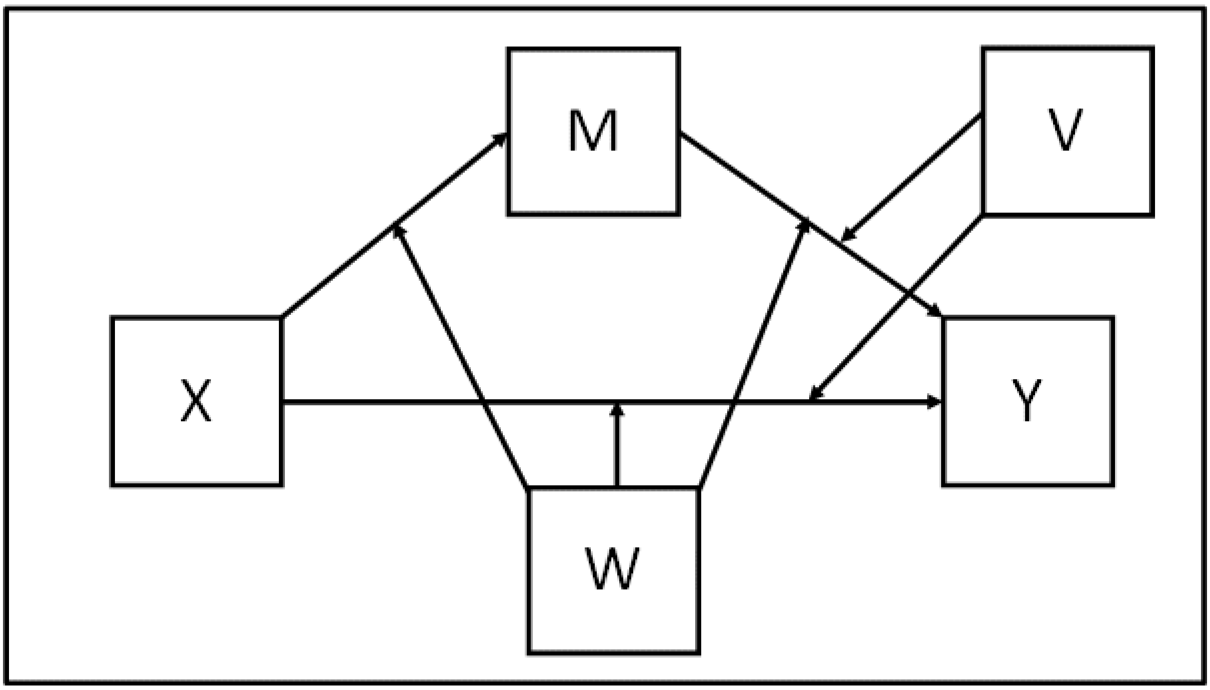
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

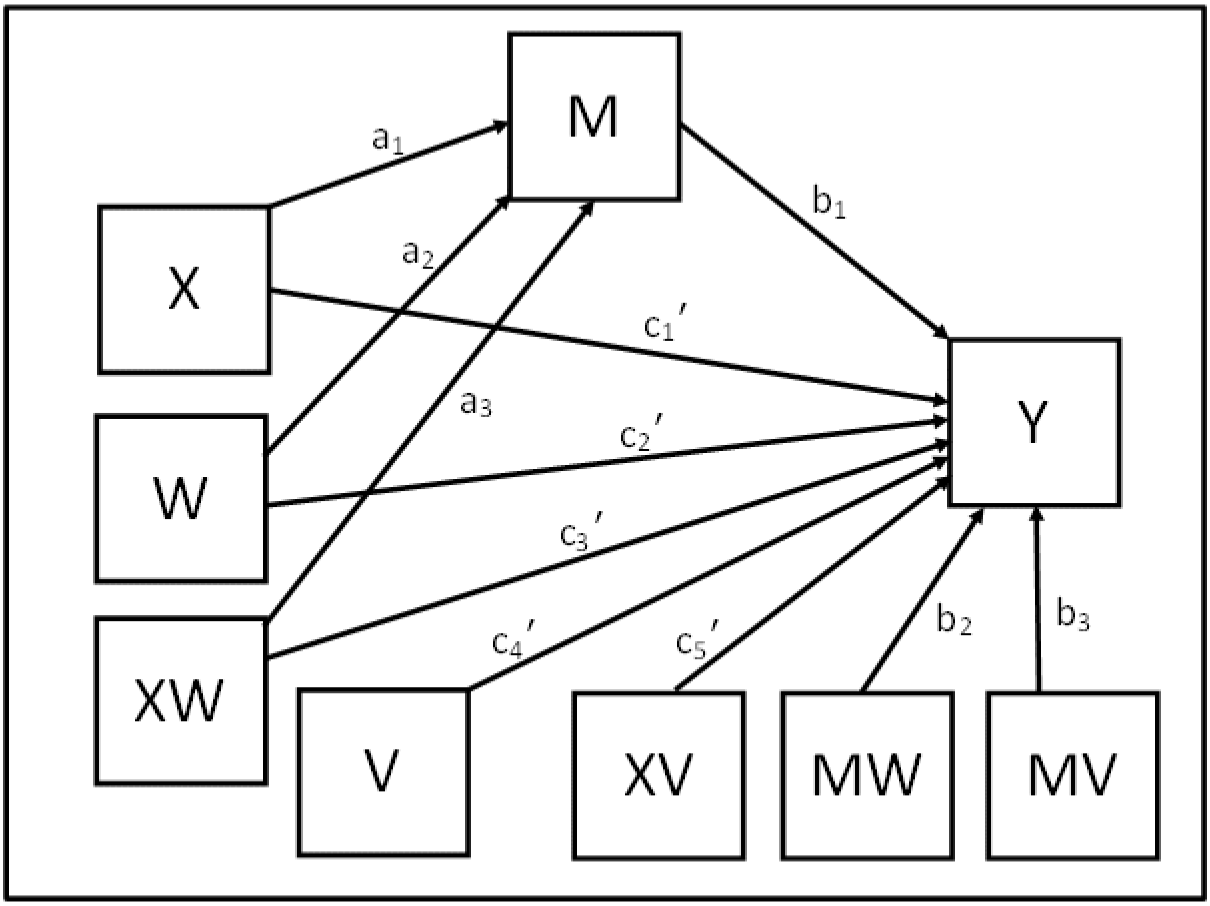
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MW + b3MV + c1'X + c2'W + c3'XW + c4'V + c5'XV
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MW + b3MV + c1'X + c2'W + c3'XW + c4'V + c5'XV
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2(a0 + a1X + a2W + a3XW)W + b3(a0 + a1X + a2W + a3XW)V + c1'X + c2'W + c3'XW + c4'V + c5'XV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + a0b2W + a1b2XW + a2b2WW + a3b2XWW + a0b3V + a1b3XV + a2b3WV + a3b3XWV + c1'X + c2'W + c3'XW + c4'V + c5'XV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a0b2W + a2b2WW + a0b3V + a2b3WV + c2'W + c4'V) + (a1b1 + a3b1W + a1b2W + a3b2WW + a1b3V + a3b3WV + c1' + c3'W + c5'V)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b2W + a3b2WW + a1b3V + a3b3WV = (a1 + a3W)(b1 + b2W + b3V)

One direct effect of X on Y, conditional on W, V:

c1' + c3'W + c5'V

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW XV MW MV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MV = M*V;
   XW = X*W;
   XV = X*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
```

```
    Y ON MW (b2);
    Y ON MV (b3);

    Y ON X(cdash1);
    Y ON W (cdash2);
    Y ON XW (cdash3);
    Y ON V (cdash4);
    Y ON XV (cdash5);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
    ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
    ILOW_HIV IMEW_HIV IHIW_HIV
    DLOW_LOV DMEW_LOV DHIW_LOV DLOW_MEV DMEW_MEV DHIW_MEV
    DLOW_HIV DMEW_HIV DHIW_HIV
    TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
    TLOW_HIV TMEW_HIV THIW_HIV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values
```

```
    ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_W +
a3*b2*LOW_W*LOW_W +
     a1*b3*LOW_V + a3*b3*LOW_W*LOW_V;
    IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b2*MED_W +
a3*b2*MED_W*MED_W +
     a1*b3*LOW_V + a3*b3*MED_W*LOW_V;
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_W +
a3*b2*HIGH_W*HIGH_W +
     a1*b3*LOW_V + a3*b3*HIGH_W*LOW_V;

    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_W +
a3*b2*LOW_W*LOW_W +
     a1*b3*MED_V + a3*b3*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b2*MED_W +
a3*b2*MED_W*MED_W +
     a1*b3*MED_V + a3*b3*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_W +
a3*b2*HIGH_W*HIGH_W +
     a1*b3*MED_V + a3*b3*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_W +
a3*b2*LOW_W*LOW_W +
     a1*b3*HIGH_V + a3*b3*LOW_W*HIGH_V;
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b2*MED_W +
a3*b2*MED_W*MED_W +
     a1*b3*HIGH_V + a3*b3*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_W +
a3*b2*HIGH_W*HIGH_W +
     a1*b3*HIGH_V + a3*b3*HIGH_W*HIGH_V;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOV = cdash1 + cdash3*LOW_W + cdash5*LOW_V;
    DMEW_LOV = cdash1 + cdash3*MED_W + cdash5*LOW_V;
    DHIW_LOV = cdash1 + cdash3*HIGH_W + cdash5*LOW_V;

    DLOW_MEV = cdash1 + cdash3*LOW_W + cdash5*MED_V;
    DMEW_MEV = cdash1 + cdash3*MED_W + cdash5*MED_V;
    DHIW_MEV = cdash1 + cdash3*HIGH_W + cdash5*MED_V;

    DLOW_HIV = cdash1 + cdash3*LOW_W + cdash5*HIGH_V;
    DMEW_HIV = cdash1 + cdash3*MED_W + cdash5*HIGH_V;
    DHIW_HIV = cdash1 + cdash3*HIGH_W + cdash5*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + DLOW_LOV;
    TMEW_LOV = IMEW_LOV + DMEW_LOV;
    THIW_LOV = IHIW_LOV + DHIW_LOV;
```

```
    TLOW_MEV = ILOW_MEV + DLOW_MEV;
    TMEW_MEV = IMEW_MEV + DMEW_MEV;
    THIW_MEV = IHIW_MEV + DHIW_MEV;

    TLOW_HIV = ILOW_HIV + DLOW_HIV;
    TMEW_HIV = IMEW_HIV + DMEW_HIV;
    THIW_HIV = IHIW_HIV + DHIW_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;

    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 68: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating the IV- Mediator path with all 2-way and 3-way interactions, 1 of which also moderates the Mediator-DV path

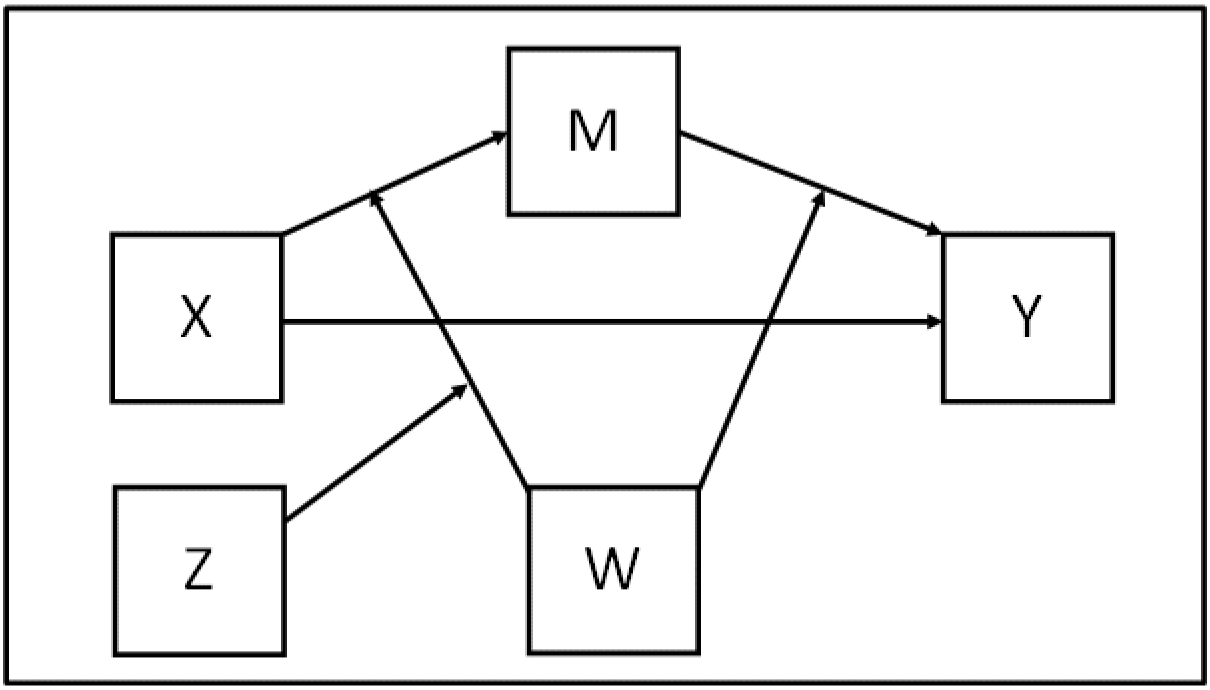
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

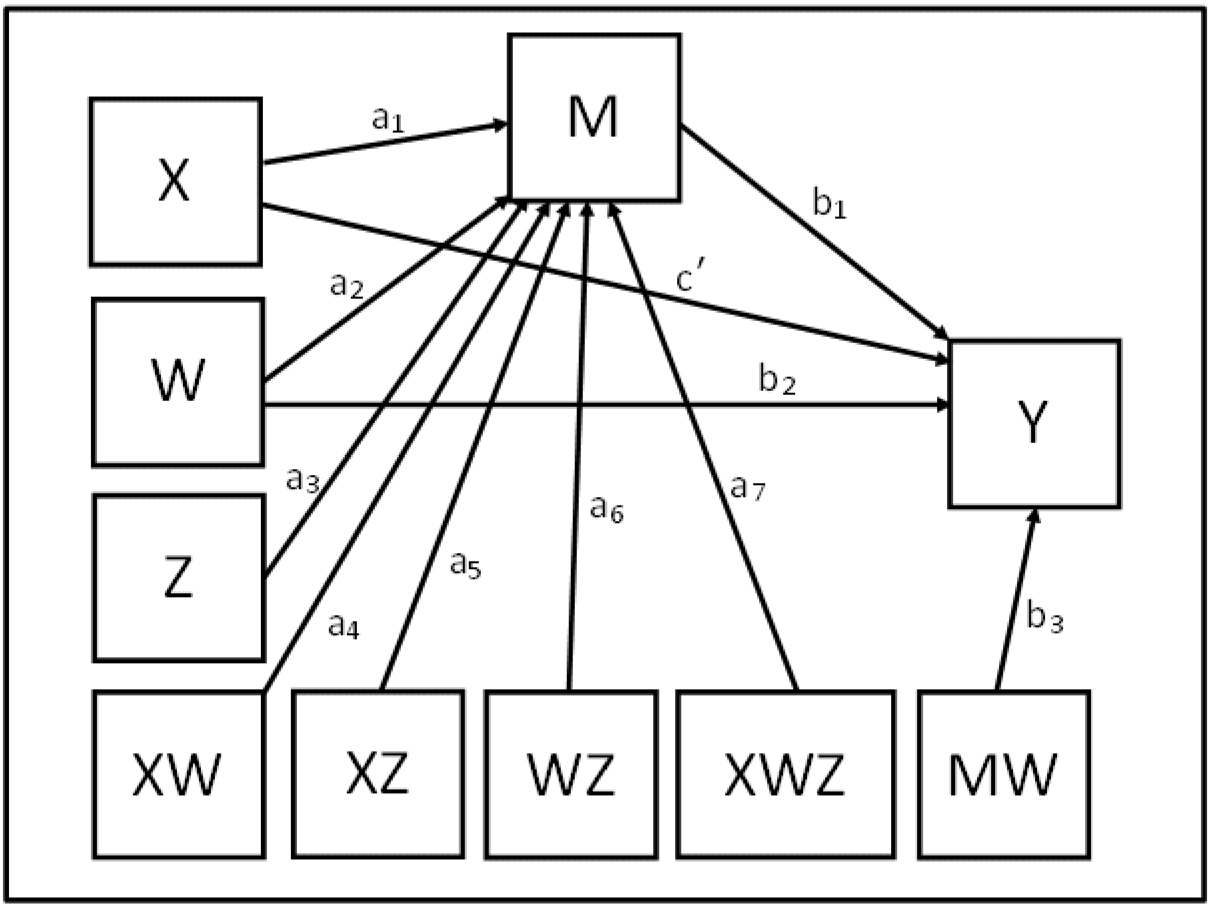
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2W + b3MW + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2W + b3MW + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2W + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)W + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + b2W + a0b3W + a1b3XW + a2b3WW + a3b3ZW + a4b3XWW + a5b3XZW + a6b3WWZ + a7b3XWWZ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + b2W + a0b3W + a2b3WW + a3b3ZW + a6b3WWZ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b3W + a4b3WW + a5b3ZW + a7b3WWZ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b3W + a4b3WW + a5b3ZW + a7b3WWZ = (a1 + a4W + a5Z + a7WZ)(b1 + b3W)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ WZ MW XWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON W (b2);
   Y ON MW (b3);

   Y ON X(cdash);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values
```

```
    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b3*LOW_W + a4*b3*LOW_W*LOW_W + a5*b3*LOW_Z*LOW_W +
     a7*b3*LOW_W*LOW_W*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b3*MED_W + a4*b3*MED_W*MED_W + a5*b3*LOW_Z*MED_W +
     a7*b3*MED_W*MED_W*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b3*HIGH_W + a4*b3*HIGH_W*HIGH_W + a5*b3*LOW_Z*HIGH_W
+
     a7*b3*HIGH_W*HIGH_W*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b3*LOW_W + a4*b3*LOW_W*LOW_W + a5*b3*MED_Z*LOW_W +
     a7*b3*LOW_W*LOW_W*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b3*MED_W + a4*b3*MED_W*MED_W + a5*b3*MED_Z*MED_W +
     a7*b3*MED_W*MED_W*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b3*HIGH_W + a4*b3*HIGH_W*HIGH_W + a5*b3*MED_Z*HIGH_W
+
     a7*b3*HIGH_W*HIGH_W*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b3*LOW_W + a4*b3*LOW_W*LOW_W + a5*b3*HIGH_Z*LOW_W +
     a7*b3*LOW_W*LOW_W*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b3*MED_W + a4*b3*MED_W*MED_W + a5*b3*HIGH_Z*MED_W +
     a7*b3*MED_W*MED_W*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b3*HIGH_W + a4*b3*HIGH_W*HIGH_W + a5*b3*HIGH_Z*HIGH_W
+
     a7*b3*HIGH_W*HIGH_W*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + cdash;
    TMEW_LOZ = IMEW_LOZ + cdash;
    THIW_LOZ = IHIW_LOZ + cdash;
```

```
    TLOW_MEZ = ILOW_MEZ + cdash;
    TMEW_MEZ = IMEW_MEZ + cdash;
    THIW_MEZ = IHIW_MEZ + cdash;

    TLOW_HIZ = ILOW_HIZ + cdash;
    TMEW_HIZ = IMEW_HIZ + cdash;
    THIW_HIZ = IHIW_HIZ + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 69: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both of the IV-Mediator path and the direct IV-DV path, with all 2-way and 3-way interactions, 1 of which also moderates the Mediator-DV path

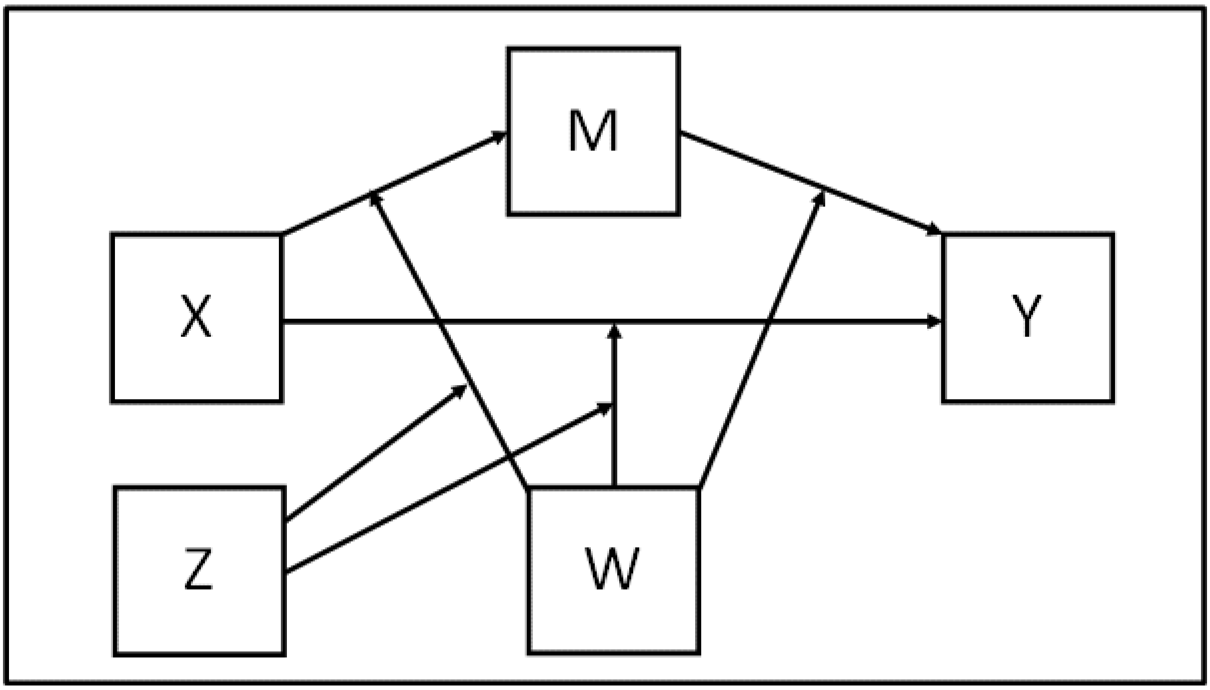
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

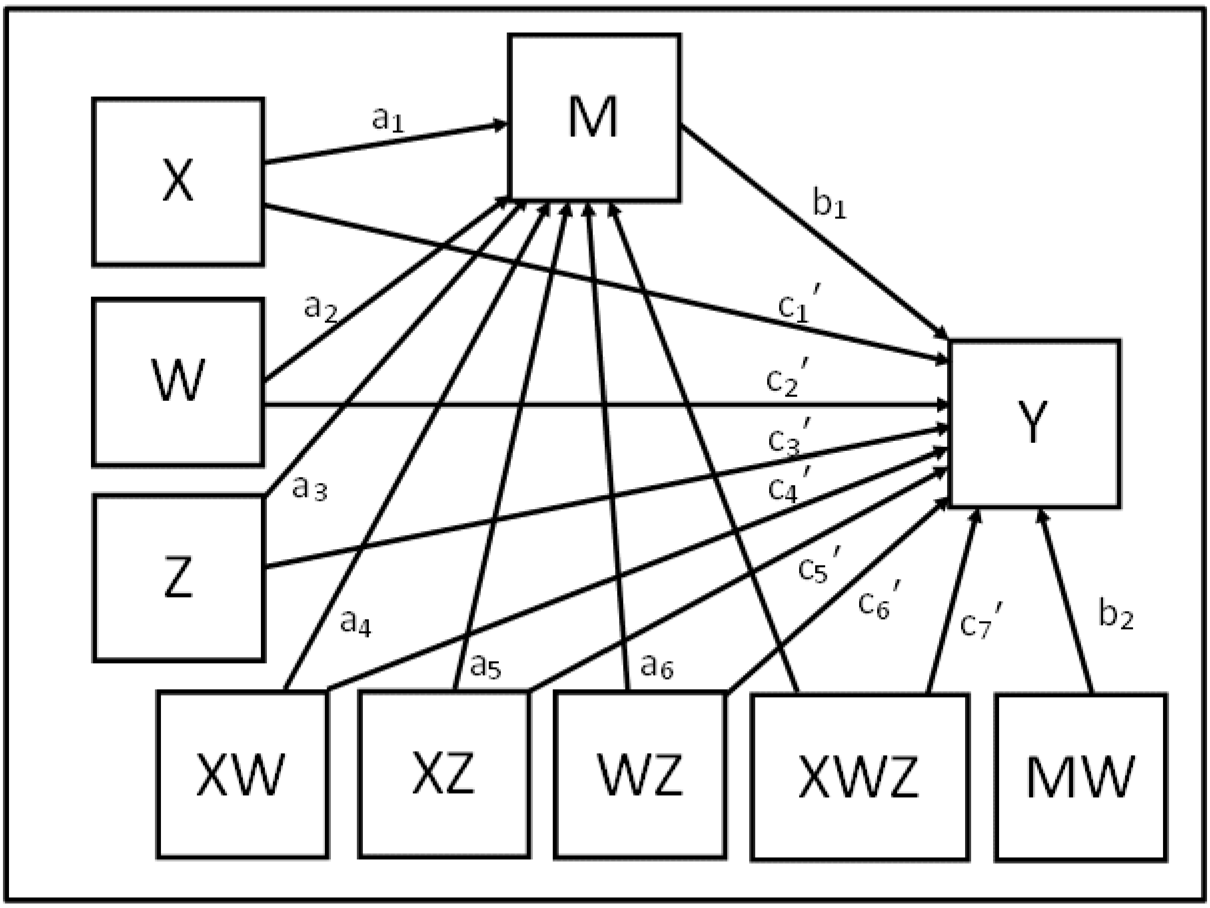
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

$Y = b0 + b1M + b2MW + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ$
$M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ$

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

$Y = b0 + b1M + b2MW + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ$
$M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ$

Hence... substituting in equation for M

$Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)W + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ$

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + a0b2W + a1b2XW + a2b2WW + a3b2ZW + a4b2XWW + a5b2XZW + a6b2WWZ + a7b2XWWZ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + a0b2W + a2b2WW + a3b2ZW + a6b2WWZ + c2'W + c3'Z + c6'WZ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2W + a4b2WW + a5b2ZW + a7b2WWZ + c1' + c4'W + c5'Z + c7'WZ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2W + a4b2WW + a5b2ZW + a7b2WWZ = (a1 + a4W + a5Z + a7WZ)(b1 + b2W)

One direct effect of X on Y, conditional on W, Z:

c1' + c4'W + c5'Z + c7'WZ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ WZ XWZ MW;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   XWZ = X*W*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON MW (b2);

    Y ON X(cdash1);
    Y ON W (cdash2);
    Y ON Z (cdash3);
    Y ON XW (cdash4);
    Y ON XZ (cdash5);
    Y ON WZ (cdash6);
    Y ON XWZ (cdash7);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
    M ON WZ (a6);
    M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W
```

```
    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_W + a4*b2*LOW_W*LOW_W + a5*b2*LOW_Z*LOW_W +
     a7*b2*LOW_W*LOW_W*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_W + a4*b2*MED_W*MED_W + a5*b2*LOW_Z*MED_W +
     a7*b2*MED_W*MED_W*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_W + a4*b2*HIGH_W*HIGH_W + a5*b2*LOW_Z*HIGH_W
+
     a7*b2*HIGH_W*HIGH_W*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_W + a4*b2*LOW_W*LOW_W + a5*b2*MED_Z*LOW_W +
     a7*b2*LOW_W*LOW_W*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_W + a4*b2*MED_W*MED_W + a5*b2*MED_Z*MED_W +
     a7*b2*MED_W*MED_W*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_W + a4*b2*HIGH_W*HIGH_W + a5*b2*MED_Z*HIGH_W
+
     a7*b2*HIGH_W*HIGH_W*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_W + a4*b2*LOW_W*LOW_W + a5*b2*HIGH_Z*LOW_W +
     a7*b2*LOW_W*LOW_W*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_W + a4*b2*MED_W*MED_W + a5*b2*HIGH_Z*MED_W +
     a7*b2*MED_W*MED_W*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_W + a4*b2*HIGH_W*HIGH_W + a5*b2*HIGH_Z*HIGH_W
```

```
+
     a7*b2*HIGH_W*HIGH_W*HIGH_Z;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + DLOW_LOZ;
    TMEW_LOZ = IMEW_LOZ + DMEW_LOZ;
    THIW_LOZ = IHIW_LOZ + DHIW_LOZ;

    TLOW_MEZ = ILOW_MEZ + DLOW_MEZ;
    TMEW_MEZ = IMEW_MEZ + DMEW_MEZ;
    THIW_MEZ = IHIW_MEZ + DHIW_MEZ;

    TLOW_HIZ = ILOW_HIZ + DLOW_HIZ;
    TMEW_HIZ = IMEW_HIZ + DMEW_HIZ;
    THIW_HIZ = IHIW_HIZ + DHIW_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis
```

```
    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 70: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating the Mediator- DV path with all 2-way and 3-way interactions, 1 of which also moderates the IV-Mediator path

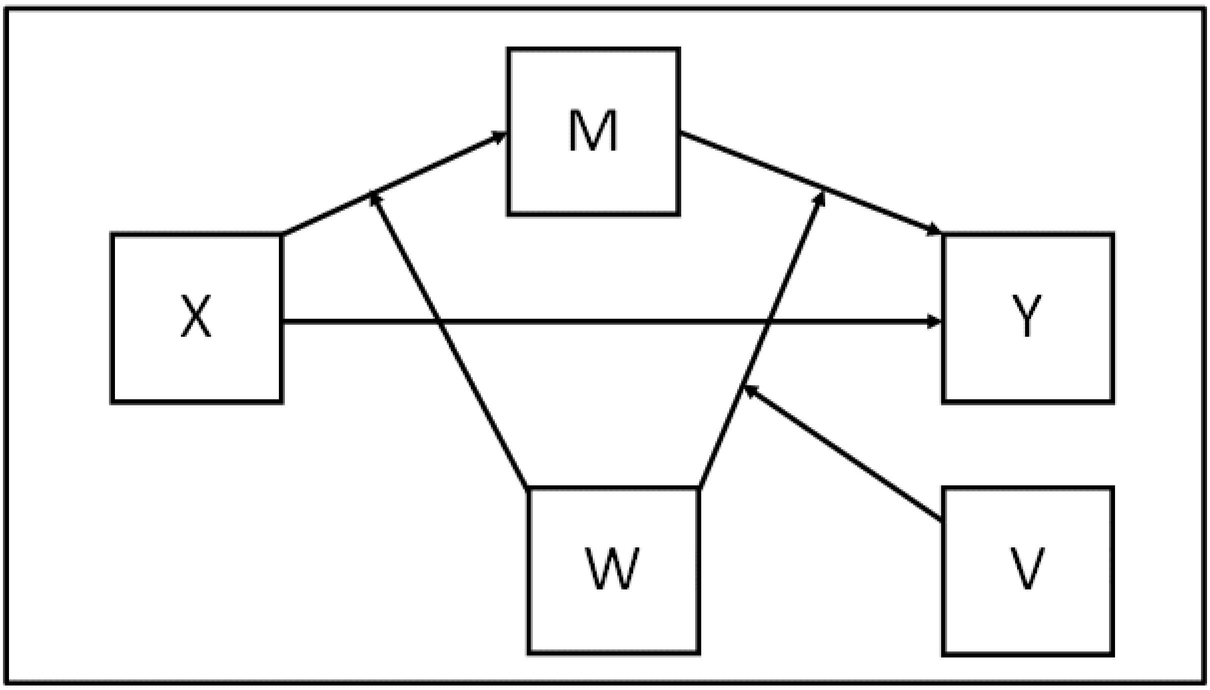
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

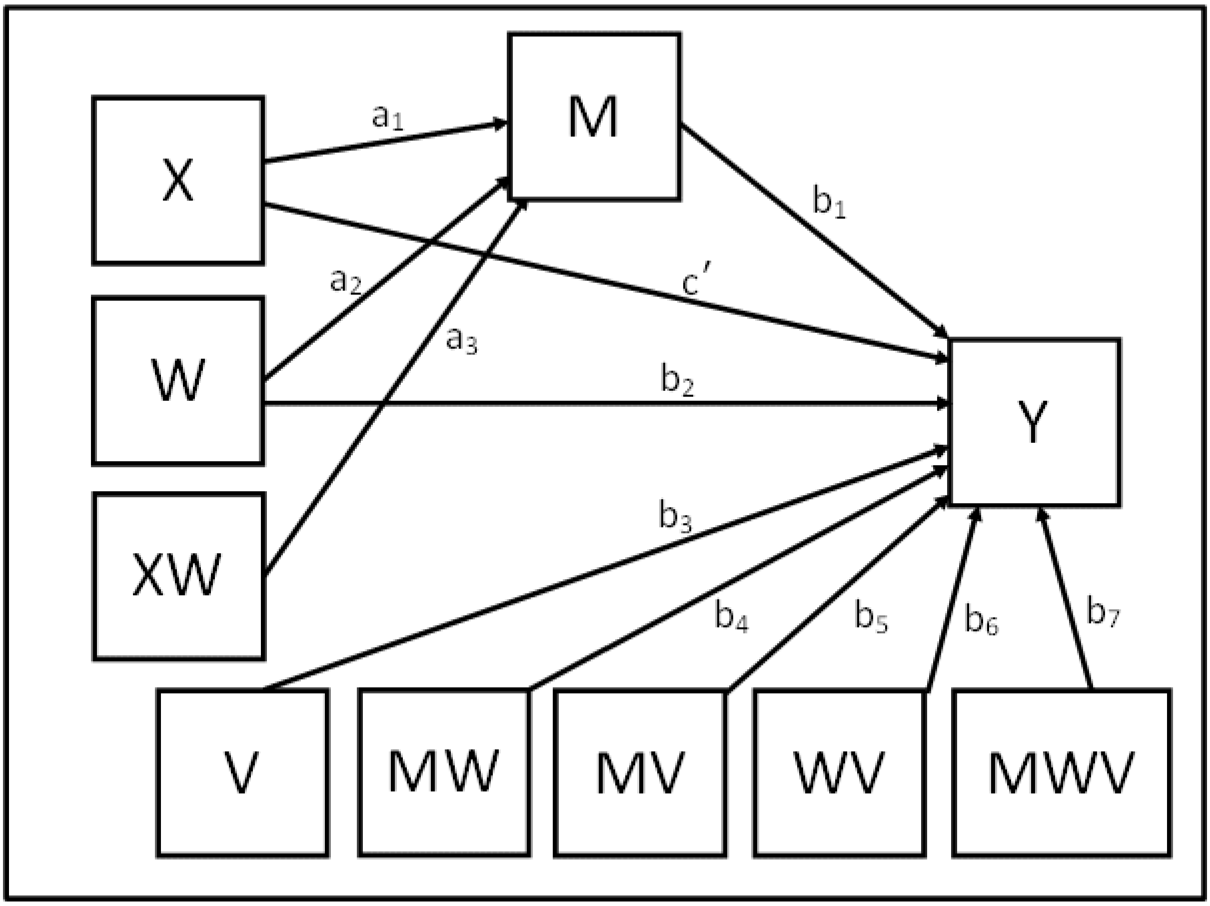
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2W + b3V + b4MW + b5MV + b6WV + b7MWV + c'X
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2W + b3V + b4MW + b5MV + b6WV + b7MWV + c'X
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2W + b3V + b4(a0 + a1X + a2W + a3XW)W + b5(a0 + a1X + a2W + a3XW)V + b6WV + b7(a0 + a1X + a2W + a3XW)WV + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + b2W + b3V + a0b4W + a1b4XW + a2b4WW + a3b4XWW + a0b5V + a1b5XV + a2b5WV + a3b5XWV + b6WV + a0b7WV + a1b7XWV + a2b7WWV + a3b7XWWV + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + b2W + b3V + a0b4W + a2b4WW + a0b5V + a2b5WV + b6WV + a0b7WV + a2b7WWV) + (a1b1 + a3b1W + a1b4W + a3b4WW + a1b5V + a3b5WV + a1b7WV + a3b7WWV + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b4W + a3b4WW + a1b5V + a3b5WV + a1b7WV + a3b7WWV = (a1 + a3W)(b1 + b4W + b5V + b7WV)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW WV MW MV MWV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MV = M*V;
   XW = X*W;
   WV = W*V;
   MWV = M*W*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses
```

```
MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON W (b2);
   Y ON V (b3);
   Y ON MW (b4);
   Y ON MV (b5);
   Y ON WV (b6);
   Y ON MWV (b7);

   Y ON X(cdash);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
    ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
    ILOW_HIV IMEW_HIV IHIW_HIV
    TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
    TLOW_HIV TMEW_HIV THIW_HIV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_V = #LOWV;   ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;   ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;   ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values
```

```
    ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_W +
a3*b4*LOW_W*LOW_W +
     a1*b5*LOW_V + a3*b5*LOW_W*LOW_V + a1*b7*LOW_W*LOW_V +
     a3*b7*LOW_W*LOW_W*LOW_V;
    IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b4*MED_W +
a3*b4*MED_W*MED_W +
     a1*b5*LOW_V + a3*b5*MED_W*LOW_V + a1*b7*MED_W*LOW_V +
     a3*b7*MED_W*MED_W*LOW_V;
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_W +
a3*b4*HIGH_W*HIGH_W +
     a1*b5*LOW_V + a3*b5*HIGH_W*LOW_V + a1*b7*HIGH_W*LOW_V +
     a3*b7*HIGH_W*HIGH_W*LOW_V;

    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_W +
a3*b4*LOW_W*LOW_W +
     a1*b5*MED_V + a3*b5*LOW_W*MED_V + a1*b7*LOW_W*MED_V +
     a3*b7*LOW_W*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b4*MED_W +
a3*b4*MED_W*MED_W +
     a1*b5*MED_V + a3*b5*MED_W*MED_V + a1*b7*MED_W*MED_V +
     a3*b7*MED_W*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_W +
a3*b4*HIGH_W*HIGH_W +
     a1*b5*MED_V + a3*b5*HIGH_W*MED_V + a1*b7*HIGH_W*MED_V +
     a3*b7*HIGH_W*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b4*LOW_W +
a3*b4*LOW_W*LOW_W +
     a1*b5*HIGH_V + a3*b5*LOW_W*HIGH_V + a1*b7*LOW_W*HIGH_V +
     a3*b7*LOW_W*LOW_W*HIGH_V;
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b4*MED_W +
a3*b4*MED_W*MED_W +
     a1*b5*HIGH_V + a3*b5*MED_W*HIGH_V + a1*b7*MED_W*HIGH_V +
     a3*b7*MED_W*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b4*HIGH_W +
a3*b4*HIGH_W*HIGH_W +
     a1*b5*HIGH_V + a3*b5*HIGH_W*HIGH_V + a1*b7*HIGH_W*HIGH_V
+
     a3*b7*HIGH_W*HIGH_W*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + cdash;
    TMEW_LOV = IMEW_LOV + cdash;
    THIW_LOV = IHIW_LOV + cdash;
```

```
    TLOW_MEV = ILOW_MEV + cdash;
    TMEW_MEV = IMEW_MEV + cdash;
    THIW_MEV = IHIW_MEV + cdash;

    TLOW_HIV = ILOW_HIV + cdash;
    TMEW_HIV = IMEW_HIV + cdash;
    THIW_HIV = IHIW_HIV + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;

    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 71: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both of the Mediator-DV path and the direct IV-DV path, with all 2-way and 3-way interactions, 1 of which also moderates the IV-Mediator path

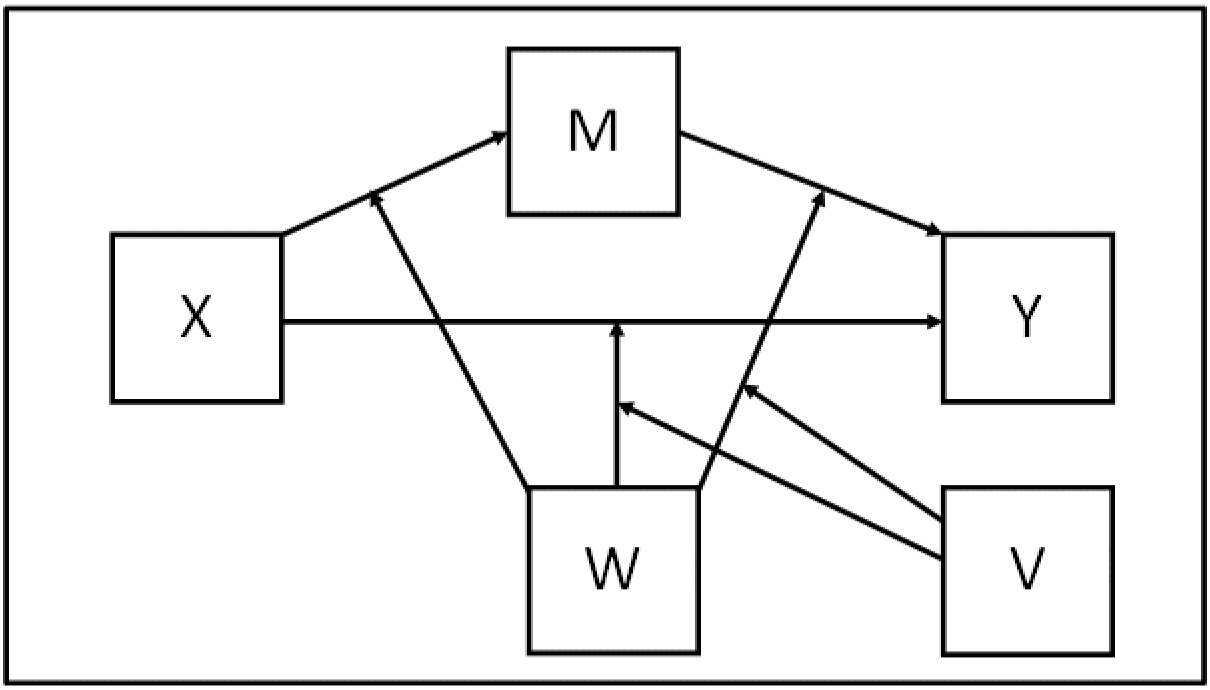
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, V, 1 outcome Y

Preliminary notes:

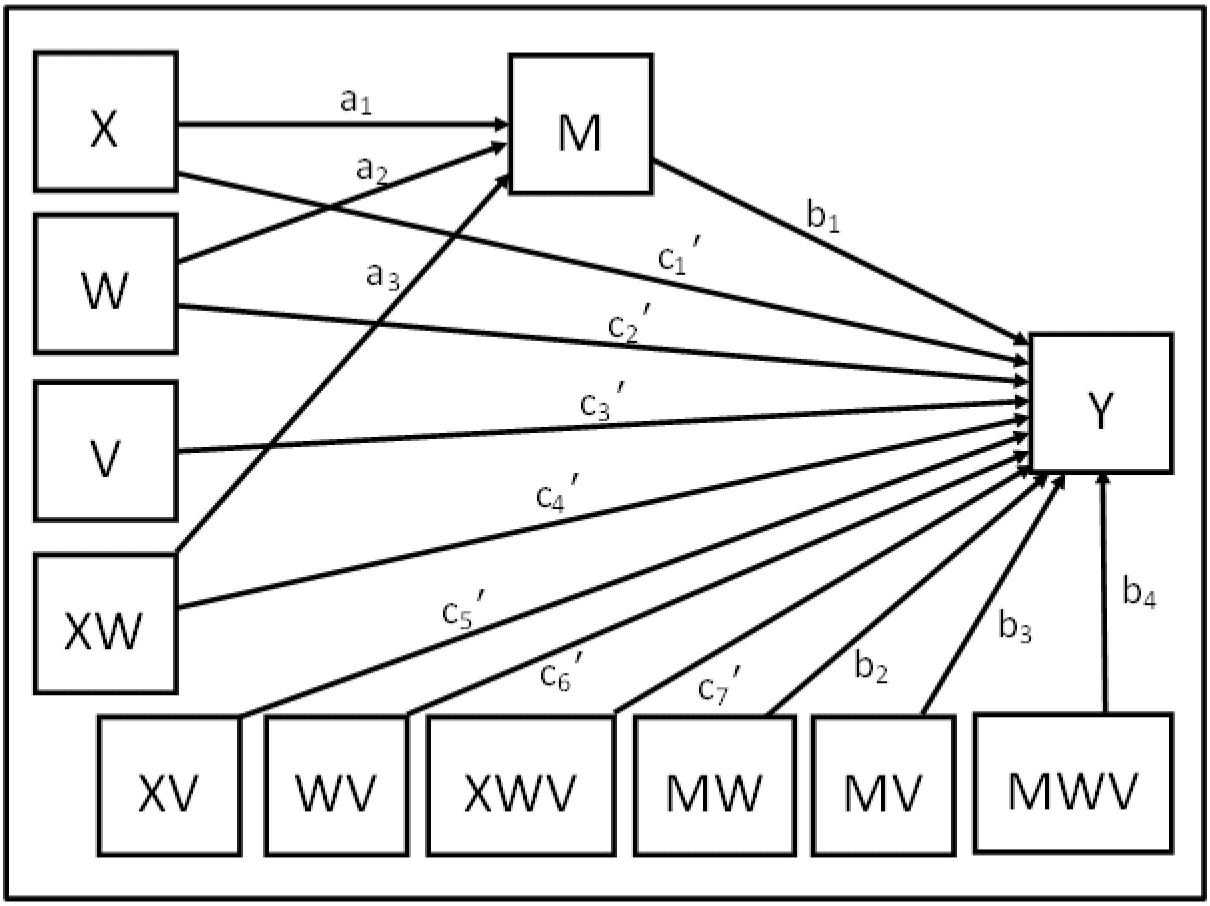
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MW + b3MV + b4MWV + c1'X + c2'W + c3'V + c4'XW + c5'XV + c6'WV + c7'XWV
M = a0 + a1X + a2W + a3XW

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MW + b3MV + b4MWV + c1'X + c2'W + c3'V + c4'XW + c5'XV + c6'WV + c7'XWV
M = a0 + a1X + a2W + a3XW

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3XW) + b2(a0 + a1X + a2W + a3XW)W + b3(a0 + a1X + a2W + a3XW)V + b4(a0 + a1X + a2W + a3XW)WV + c1'X + c2'W + c3'V + c4'XW + c5'XV + c6'WV + c7'XWV

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1XW + a0b2W + a1b2XW + a2b2WW + a3b2XWW + a0b3V + a1b3XV + a2b3WV + a3b3XWV + a0b4WV + a1b4XWV + a2b4WWV + a3b4XWWV + c1'X + c2'W + c3'V + c4'XW + c5'XV + c6'WV + c7'XWV

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a0b2W + a2b2WW + a0b3V + a2b3WV + a0b4WV + a2b4WWV + c2'W + c3'V + c6'WV) + (a1b1 + a3b1W + a1b2W + a3b2WW + a1b3V + a3b3WV + a1b4WV + a3b4WWV + c1' + c4'W + c5'V + c7'WV)X

Hence...

One indirect effect(s) of X on Y, conditional on W, V:

a1b1 + a3b1W + a1b2W + a3b2WW + a1b3V + a3b3WV + a1b4WV + a3b4WWV = (a1 + a3W)(b1 + b2W + b3V + b4WV)

One direct effect of X on Y, conditional on W, V:

c1' + c4'W + c5'V + c7'WV

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, V
! Outcome variable - Y

USEVARIABLES = X M W V Y XW XV WV MW MV XWV MWV;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MV = M*V;
   XW = X*W;
   XV = X*V;
   WV = W*V;
   MWV = M*W*V;
   XWV = X*W*V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;
```

```
! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON MW (b2);
   Y ON MV (b3);
   Y ON MWV (b4);

   Y ON X(cdash1);
   Y ON W (cdash2);
   Y ON V (cdash3);
   Y ON XW (cdash4);
   Y ON XV (cdash5);
   Y ON WV (cdash6);
   Y ON XWV (cdash7);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON XW (a3);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
V
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
    ILOW_LOV IMEW_LOV IHIW_LOV ILOW_MEV IMEW_MEV IHIW_MEV
    ILOW_HIV IMEW_HIV IHIW_HIV
    DLOW_LOV DMEW_LOV DHIW_LOV DLOW_MEV DMEW_MEV DHIW_MEV
    DLOW_HIV DMEW_HIV DHIW_HIV
    TLOW_LOV TMEW_LOV THIW_LOV TLOW_MEV TMEW_MEV THIW_MEV
    TLOW_HIV TMEW_HIV THIW_HIV);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W
```

```
    LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
    MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
    HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOV = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_W +
a3*b2*LOW_W*LOW_W +
     a1*b3*LOW_V + a3*b3*LOW_W*LOW_V + a1*b4*LOW_W*LOW_V +
     a3*b4*LOW_W*LOW_W*LOW_V;
    IMEW_LOV = a1*b1 + a3*b1*MED_W + a1*b2*MED_W +
a3*b2*MED_W*MED_W +
     a1*b3*LOW_V + a3*b3*MED_W*LOW_V + a1*b4*MED_W*LOW_V +
     a3*b4*MED_W*MED_W*LOW_V;
    IHIW_LOV = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_W +
a3*b2*HIGH_W*HIGH_W +
     a1*b3*LOW_V + a3*b3*HIGH_W*LOW_V + a1*b4*HIGH_W*LOW_V +
     a3*b4*HIGH_W*HIGH_W*LOW_V;

    ILOW_MEV = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_W +
a3*b2*LOW_W*LOW_W +
     a1*b3*MED_V + a3*b3*LOW_W*MED_V + a1*b4*LOW_W*MED_V +
     a3*b4*LOW_W*LOW_W*MED_V;
    IMEW_MEV = a1*b1 + a3*b1*MED_W + a1*b2*MED_W +
a3*b2*MED_W*MED_W +
     a1*b3*MED_V + a3*b3*MED_W*MED_V + a1*b4*MED_W*MED_V +
     a3*b4*MED_W*MED_W*MED_V;
    IHIW_MEV = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_W +
a3*b2*HIGH_W*HIGH_W +
     a1*b3*MED_V + a3*b3*HIGH_W*MED_V + a1*b4*HIGH_W*MED_V +
     a3*b4*HIGH_W*HIGH_W*MED_V;

    ILOW_HIV = a1*b1 + a3*b1*LOW_W + a1*b2*LOW_W +
a3*b2*LOW_W*LOW_W +
     a1*b3*HIGH_V + a3*b3*LOW_W*HIGH_V + a1*b4*LOW_W*HIGH_V +
     a3*b4*LOW_W*LOW_W*HIGH_V;
    IMEW_HIV = a1*b1 + a3*b1*MED_W + a1*b2*MED_W +
a3*b2*MED_W*MED_W +
     a1*b3*HIGH_V + a3*b3*MED_W*HIGH_V + a1*b4*MED_W*HIGH_V +
     a3*b4*MED_W*MED_W*HIGH_V;
    IHIW_HIV = a1*b1 + a3*b1*HIGH_W + a1*b2*HIGH_W +
a3*b2*HIGH_W*HIGH_W +
     a1*b3*HIGH_V + a3*b3*HIGH_W*HIGH_V + a1*b4*HIGH_W*HIGH_V
+
     a3*b4*HIGH_W*HIGH_W*HIGH_V;
```

```
! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOV = cdash1 + cdash4*LOW_W + cdash5*LOW_V +
cdash7*LOW_W*LOW_V;
    DMEW_LOV = cdash1 + cdash4*MED_W + cdash5*LOW_V +
cdash7*MED_W*LOW_V;
    DHIW_LOV = cdash1 + cdash4*HIGH_W + cdash5*LOW_V +
cdash7*HIGH_W*LOW_V;

    DLOW_MEV = cdash1 + cdash4*LOW_W + cdash5*MED_V +
cdash7*LOW_W*MED_V;
    DMEW_MEV = cdash1 + cdash4*MED_W + cdash5*MED_V +
cdash7*MED_W*MED_V;
    DHIW_MEV = cdash1 + cdash4*HIGH_W + cdash5*MED_V +
cdash7*HIGH_W*MED_V;

    DLOW_HIV = cdash1 + cdash4*LOW_W + cdash5*HIGH_V +
cdash7*LOW_W*HIGH_V;
    DMEW_HIV = cdash1 + cdash4*MED_W + cdash5*HIGH_V +
cdash7*MED_W*HIGH_V;
    DHIW_HIV = cdash1 + cdash4*HIGH_W + cdash5*HIGH_V +
cdash7*HIGH_W*HIGH_V;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOV = ILOW_LOV + DLOW_LOV;
    TMEW_LOV = IMEW_LOV + DMEW_LOV;
    THIW_LOV = IHIW_LOV + DHIW_LOV;

    TLOW_MEV = ILOW_MEV + DLOW_MEV;
    TMEW_MEV = IMEW_MEV + DMEW_MEV;
    THIW_MEV = IHIW_MEV + DHIW_MEV;

    TLOW_HIV = ILOW_HIV + DLOW_HIV;
    TMEW_HIV = IMEW_HIV + DMEW_HIV;
    THIW_HIV = IHIW_HIV + DHIW_HIV;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOV PMEW_LOV PHIW_LOV PLOW_MEV PMEW_MEV
PHIW_MEV
    PLOW_HIV PMEW_HIV PHIW_HIV);

    LOOP(XVAL,1,5,0.1);
```

```
    PLOW_LOV = ILOW_LOV*XVAL;
    PMEW_LOV = IMEW_LOV*XVAL;
    PHIW_LOV = IHIW_LOV*XVAL;

    PLOW_MEV = ILOW_MEV*XVAL;
    PMEW_MEV = IMEW_MEV*XVAL;
    PHIW_MEV = IHIW_MEV*XVAL;

    PLOW_HIV = ILOW_HIV*XVAL;
    PMEW_HIV = IMEW_HIV*XVAL;
    PHIW_HIV = IHIW_HIV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 72: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV- Mediator path and the Mediator-DV path, with all 2-way and 3-way interactions

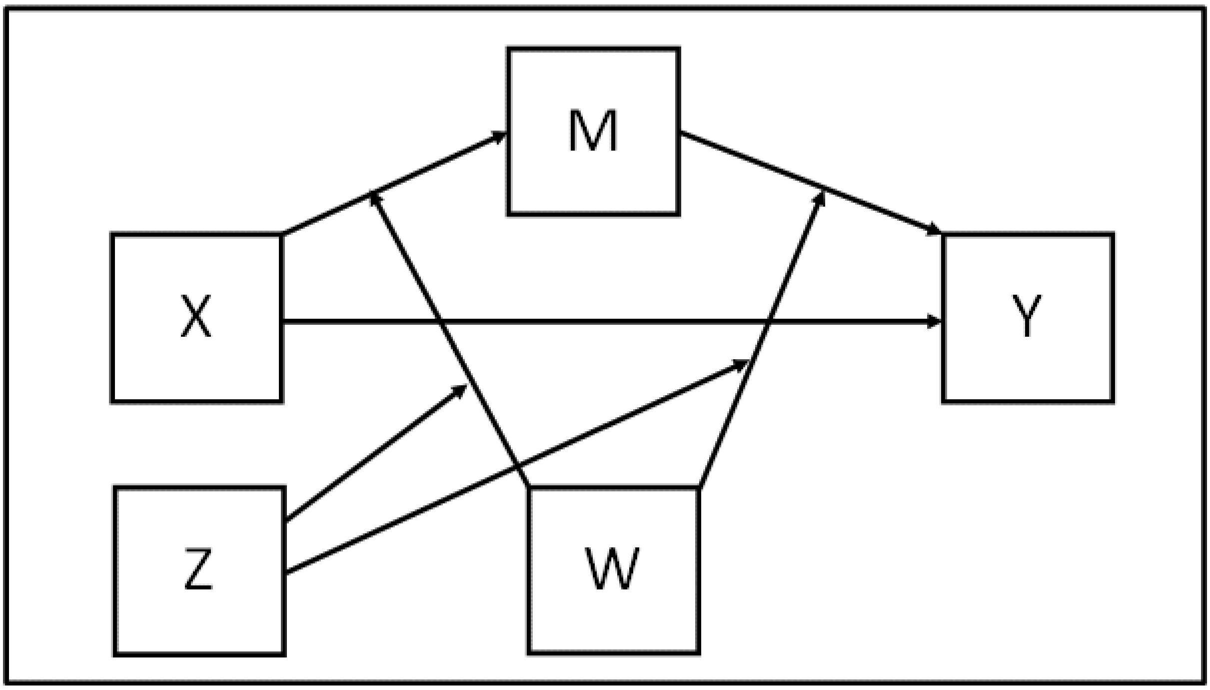
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

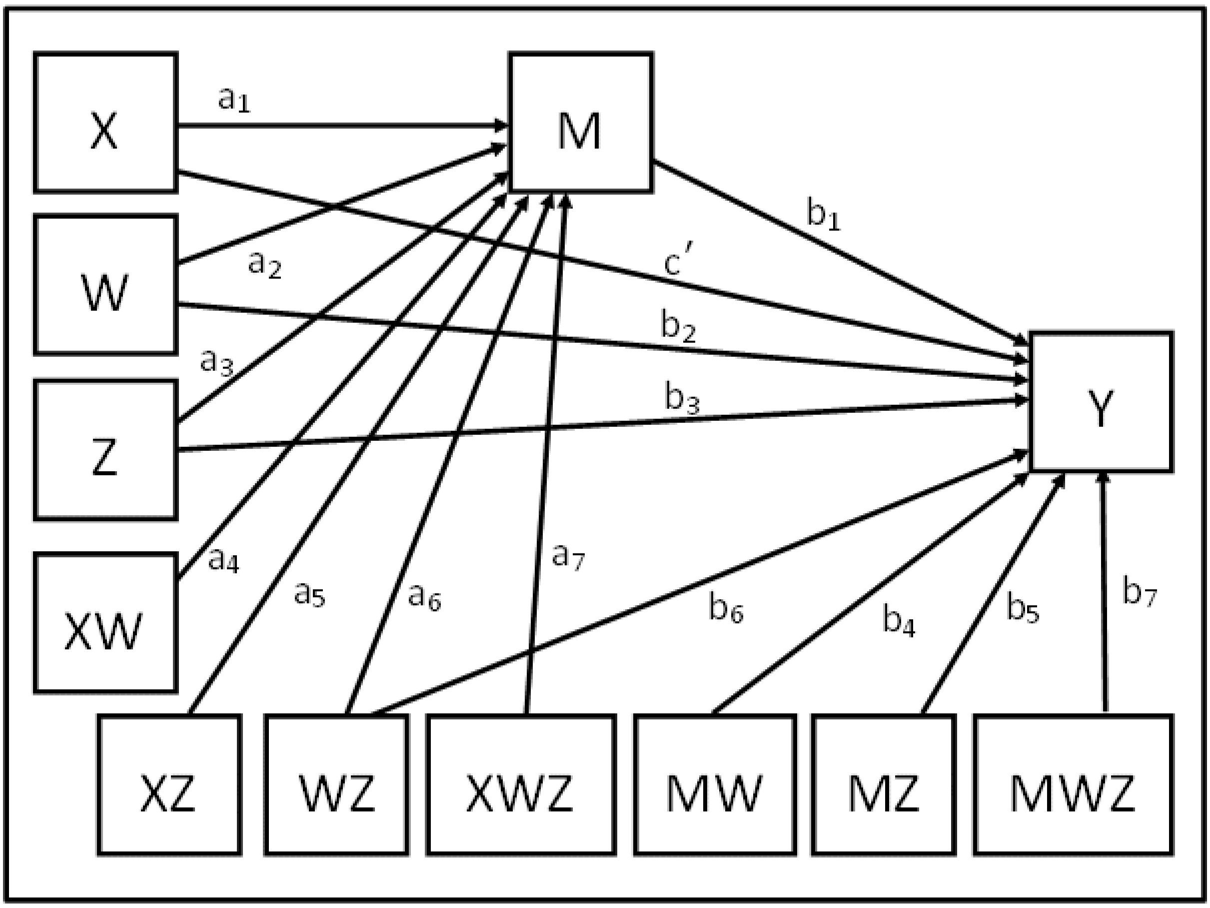
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2W + b3Z + b4MW + b5MZ + b6WZ + b7MWZ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2W + b3Z + b4MW + b5MZ + b6WZ + b7MWZ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2W + b3Z + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)W + b5(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)Z + b6WZ + b7(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)WZ + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + b2W + b3Z + a0b4W + a1b4XW + a2b4WW + a3b4ZW + a4b4XWW + a5b4XZW + a6b4WWZ + a7b4XWWZ + a0b5Z + a1b5XZ + a2b5WZ + a3b5ZZ + a4b5XWZ + a5b5XZZ + a6b5WZZ + a7b5XWZZ + b6WZ + a0b7WZ + a1b7XWZ + a2b7WWZ + a3b7WZZ + a4b7XWWZ + a5b7XWZZ + a6b7WWZZ + a7XWWZZ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + b2W + b3Z + a0b4W + a2b4WW + a3b4ZW + a6b4WWZ + a0b5Z + a2b5WZ + a3b5ZZ + a6b5WZZ + b6WZ + a0b7WZ + a2b7WWZ + a3b7WZZ + a6b7WWZZ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4W + a4b4WW + a5b4ZW + a7b4WWZ + a1b5Z + a4b5WZ + a5b5ZZ + a7b5WZZ + a1b7WZ + a4b7WWZ + a5b7WZZ + a7b7WWZZ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b4W + a4b4WW + a5b4ZW + a7b4WWZ + a1b5Z + a4b5WZ + a5b5ZZ + a7b5WZZ + a1b7WZ + a4b7WWZ + a5b7WZZ + a7b7WWZZ = (a1 + a4W + a5Z + a7WZ)(b1 + b4W + b5Z + b7WZ)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ WZ MW MZ XWZ MWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MZ = M*Z;
   XW = X*W;
   XZ = X*Z;
   WZ = W*Z;
   MWZ = M*W*Z;
   XWZ = X*W*Z;
```

```
ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON W (b2);
   Y ON Z (b3);
   Y ON MW (b4);
   Y ON MZ (b5);
   Y ON WZ (b6);
   Y ON MWZ (b7);

   Y ON X(cdash);

   [M] (a0);
   M ON X (a1);
   M ON W (a2);
   M ON Z (a3);
   M ON XW (a4);
   M ON XZ (a5);
   M ON WZ (a6);
   M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
```

```
    HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;    ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;    ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;    ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b4*LOW_W + a4*b4*LOW_W*LOW_W + a5*b4*LOW_Z*LOW_W +
     a7*b4*LOW_W*LOW_W*LOW_Z + a1*b5*LOW_Z + a4*b5*LOW_W*LOW_Z
+
     a5*b5*LOW_Z*LOW_Z + a7*b5*LOW_W*LOW_Z*LOW_Z +
a1*b7*LOW_W*LOW_Z +
     a4*b7*LOW_W*LOW_W*LOW_Z + a5*b7*LOW_W*LOW_Z*LOW_Z +
     a7*b7*LOW_W*LOW_W*LOW_Z*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b4*MED_W + a4*b4*MED_W*MED_W + a5*b4*LOW_Z*MED_W +
     a7*b4*MED_W*MED_W*LOW_Z + a1*b5*LOW_Z + a4*b5*MED_W*LOW_Z
+
     a5*b5*LOW_Z*LOW_Z + a7*b5*MED_W*LOW_Z*LOW_Z +
a1*b7*MED_W*LOW_Z +
     a4*b7*MED_W*MED_W*LOW_Z + a5*b7*MED_W*LOW_Z*LOW_Z +
     a7*b7*MED_W*MED_W*LOW_Z*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b4*HIGH_W + a4*b4*HIGH_W*HIGH_W + a5*b4*LOW_Z*HIGH_W
+
     a7*b4*HIGH_W*HIGH_W*LOW_Z + a1*b5*LOW_Z +
a4*b5*HIGH_W*LOW_Z +
     a5*b5*LOW_Z*LOW_Z + a7*b5*HIGH_W*LOW_Z*LOW_Z +
a1*b7*HIGH_W*LOW_Z +
     a4*b7*HIGH_W*HIGH_W*LOW_Z + a5*b7*HIGH_W*LOW_Z*LOW_Z +
     a7*b7*HIGH_W*HIGH_W*LOW_Z*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b4*LOW_W + a4*b4*LOW_W*LOW_W + a5*b4*MED_Z*LOW_W +
     a7*b4*LOW_W*LOW_W*MED_Z + a1*b5*MED_Z + a4*b5*LOW_W*MED_Z
+
     a5*b5*MED_Z*MED_Z + a7*b5*LOW_W*MED_Z*MED_Z +
a1*b7*LOW_W*MED_Z +
```

```
     a4*b7*LOW_W*LOW_W*MED_Z + a5*b7*LOW_W*MED_Z*MED_Z +
     a7*b7*LOW_W*LOW_W*MED_Z*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b4*MED_W + a4*b4*MED_W*MED_W + a5*b4*MED_Z*MED_W +
     a7*b4*MED_W*MED_W*MED_Z + a1*b5*MED_Z + a4*b5*MED_W*MED_Z
+
     a5*b5*MED_Z*MED_Z + a7*b5*MED_W*MED_Z*MED_Z +
a1*b7*MED_W*MED_Z +
     a4*b7*MED_W*MED_W*MED_Z + a5*b7*MED_W*MED_Z*MED_Z +
     a7*b7*MED_W*MED_W*MED_Z*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b4*HIGH_W + a4*b4*HIGH_W*HIGH_W + a5*b4*MED_Z*HIGH_W
+
     a7*b4*HIGH_W*HIGH_W*MED_Z + a1*b5*MED_Z +
a4*b5*HIGH_W*MED_Z +
     a5*b5*MED_Z*MED_Z + a7*b5*HIGH_W*MED_Z*MED_Z +
a1*b7*HIGH_W*MED_Z +
     a4*b7*HIGH_W*HIGH_W*MED_Z + a5*b7*HIGH_W*MED_Z*MED_Z +
     a7*b7*HIGH_W*HIGH_W*MED_Z*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b4*LOW_W + a4*b4*LOW_W*LOW_W + a5*b4*HIGH_Z*LOW_W +
     a7*b4*LOW_W*LOW_W*HIGH_Z + a1*b5*HIGH_Z +
a4*b5*LOW_W*HIGH_Z +
     a5*b5*HIGH_Z*HIGH_Z + a7*b5*LOW_W*HIGH_Z*HIGH_Z +
a1*b7*LOW_W*HIGH_Z +
     a4*b7*LOW_W*LOW_W*HIGH_Z + a5*b7*LOW_W*HIGH_Z*HIGH_Z +
     a7*b7*LOW_W*LOW_W*HIGH_Z*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b4*MED_W + a4*b4*MED_W*MED_W + a5*b4*HIGH_Z*MED_W +
     a7*b4*MED_W*MED_W*HIGH_Z + a1*b5*HIGH_Z +
a4*b5*MED_W*HIGH_Z +
     a5*b5*HIGH_Z*HIGH_Z + a7*b5*MED_W*HIGH_Z*HIGH_Z +
a1*b7*MED_W*HIGH_Z +
     a4*b7*MED_W*MED_W*HIGH_Z + a5*b7*MED_W*HIGH_Z*HIGH_Z +
     a7*b7*MED_W*MED_W*HIGH_Z*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b4*HIGH_W + a4*b4*HIGH_W*HIGH_W + a5*b4*HIGH_Z*HIGH_W
+
     a7*b4*HIGH_W*HIGH_W*HIGH_Z + a1*b5*HIGH_Z +
a4*b5*HIGH_W*HIGH_Z +
     a5*b5*HIGH_Z*HIGH_Z + a7*b5*HIGH_W*HIGH_Z*HIGH_Z +
a1*b7*HIGH_W*HIGH_Z +
```

```
     a4*b7*HIGH_W*HIGH_W*HIGH_Z + a5*b7*HIGH_W*HIGH_Z*HIGH_Z
+
     a7*b7*HIGH_W*HIGH_W*HIGH_Z*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + cdash;
    TMEW_LOZ = IMEW_LOZ + cdash;
    THIW_LOZ = IHIW_LOZ + cdash;

    TLOW_MEZ = ILOW_MEZ + cdash;
    TMEW_MEZ = IMEW_MEZ + cdash;
    THIW_MEZ = IHIW_MEZ + cdash;

    TLOW_HIZ = ILOW_HIZ + cdash;
    TMEW_HIZ = IMEW_HIZ + cdash;
    THIW_HIZ = IHIW_HIZ + cdash;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 73: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating each of the IV-Mediator path, the Mediator-DV path and the direct IV-DV path, with all 2-way and 3-way interactions

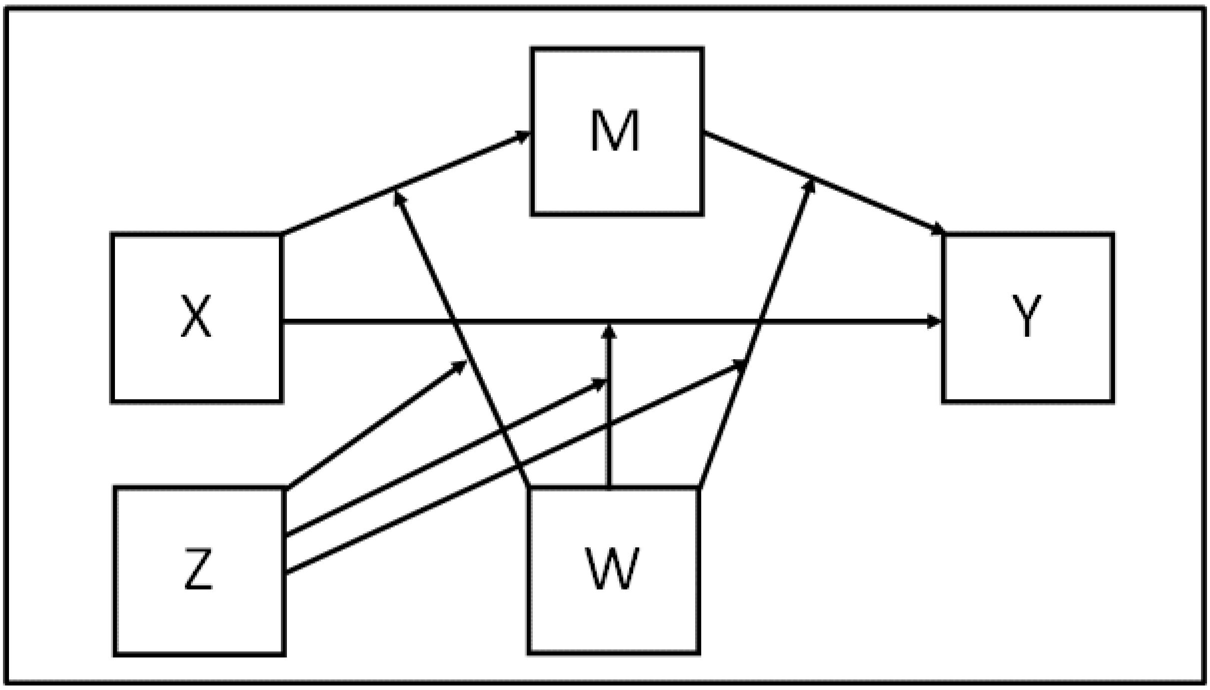
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

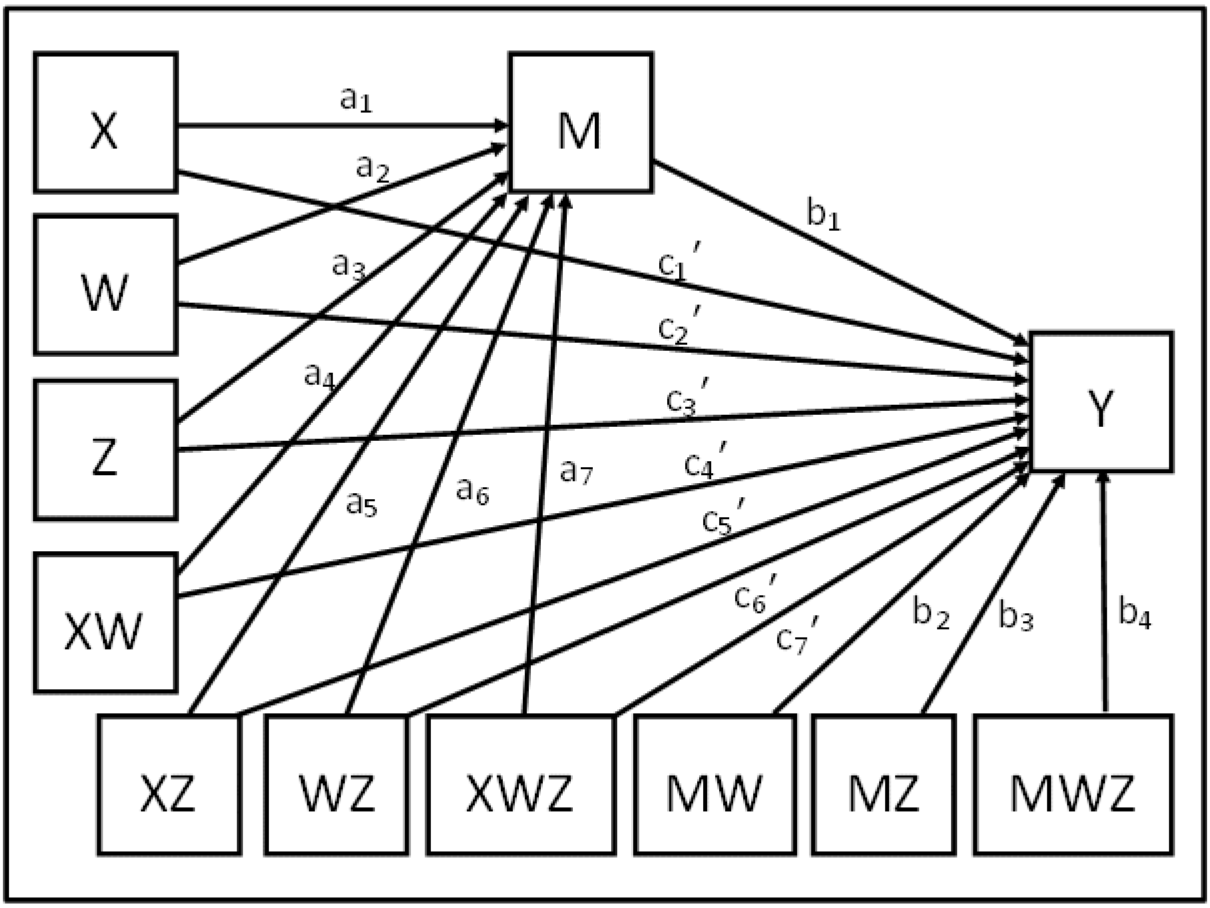
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MW + b3MZ + b4MWZ + c1'X + c2W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MW + b3MZ + b4MWZ + c1'X + c2W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)W + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)Z + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ + a6WZ + a7XWZ)WZ + c1'X + c2W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a6b1WZ + a7b1XWZ + a0b2W + a1b2XW + a2b2WW + a3b2ZW + a4b2XWW + a5b2XZW + a6b2WWZ + a7b2XWWZ + a0b3Z + a1b3XZ + a2b3WZ + a3b3ZZ + a4b3XWZ + a5b3XZZ + a6b3WZZ + a7b3XWZZ + a0b4WZ + a1b4XWZ + a2b4WWZ + a3b4WZZ + a4b4XWWZ + a5b4XWZZ + a6b4WWZZ + a7b4XWWZZ + c1'X + c2W + c3'Z + c4'XW + c5'XZ + c6'WZ + c7'XWZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a6b1WZ + a0b2W + a2b2WW + a3b2ZW + a6b2WWZ + a0b3Z + a2b3WZ + a3b3ZZ + a6b3WZZ + a0b4WZ + a2b4WWZ + a3b4WZZ + a6b4WWZZ+ c2W + c3'Z + c6'WZ) + (a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2W + a4b2WW + a5b2ZW + a7b2WWZ + a1b3Z + a4b3WZ + a5b3ZZ + a7b3WZZ + a1b4WZ + a4b4WWZ + a5b4WZZ + a7b4WWZZ + c1' + c4'W + c5'Z + c7'WZ)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a7b1WZ + a1b2W + a4b2WW + a5b2ZW + a7b2WWZ + a1b3Z + a4b3WZ + a5b3ZZ + a7b3WZZ + a1b4WZ + a4b4WWZ + a5b4WZZ + a7b4WWZZ = (a1 + a4W + a5Z + a7WZ)(b1 + b2W + b3Z + b4WZ)

One direct effect of X on Y, conditional on W, Z:

c1' + c4'W + c5'Z + c7'WZ

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ WZ MW MZ XWZ MWZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MZ = M*Z;
   XW = X*W;
   XZ = X*Z;
```

```
    WZ = W*Z;
    MWZ = M*W*Z;
    XWZ = X*W*Z;

ANALYSIS:
    TYPE = GENERAL;
    ESTIMATOR = ML;
    BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
    [Y] (b0);
    Y ON M (b1);
    Y ON MW (b2);
    Y ON MZ (b3);
    Y ON MWZ (b4);

    Y ON X(cdash1);
    Y ON W (cdash2);
    Y ON Z (cdash3);
    Y ON XW (cdash4);
    Y ON XZ (cdash5);
    Y ON WZ (cdash6);
    Y ON XWZ (cdash7);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);
    M ON WZ (a6);
    M ON XWZ (a7);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
     NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
     ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
     ILOW_HIZ IMEW_HIZ IHIW_HIZ
```

```
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z +
a7*b1*LOW_W*LOW_Z +
     a1*b2*LOW_W + a4*b2*LOW_W*LOW_W + a5*b2*LOW_Z*LOW_W +
     a7*b2*LOW_W*LOW_W*LOW_Z + a1*b3*LOW_Z + a4*b3*LOW_W*LOW_Z
+
     a5*b3*LOW_Z*LOW_Z + a7*b3*LOW_W*LOW_Z*LOW_Z +
a1*b4*LOW_W*LOW_Z +
     a4*b4*LOW_W*LOW_W*LOW_Z + a5*b4*LOW_W*LOW_Z*LOW_Z +
     a7*b4*LOW_W*LOW_W*LOW_Z*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z +
a7*b1*MED_W*LOW_Z +
     a1*b2*MED_W + a4*b2*MED_W*MED_W + a5*b2*LOW_Z*MED_W +
     a7*b2*MED_W*MED_W*LOW_Z + a1*b3*LOW_Z + a4*b3*MED_W*LOW_Z
+
     a5*b3*LOW_Z*LOW_Z + a7*b3*MED_W*LOW_Z*LOW_Z +
a1*b4*MED_W*LOW_Z +
     a4*b4*MED_W*MED_W*LOW_Z + a5*b4*MED_W*LOW_Z*LOW_Z +
     a7*b4*MED_W*MED_W*LOW_Z*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a7*b1*HIGH_W*LOW_Z +
     a1*b2*HIGH_W + a4*b2*HIGH_W*HIGH_W + a5*b2*LOW_Z*HIGH_W
+
     a7*b2*HIGH_W*HIGH_W*LOW_Z + a1*b3*LOW_Z +
a4*b3*HIGH_W*LOW_Z +
     a5*b3*LOW_Z*LOW_Z + a7*b3*HIGH_W*LOW_Z*LOW_Z +
a1*b4*HIGH_W*LOW_Z +
```

```
     a4*b4*HIGH_W*HIGH_W*LOW_Z + a5*b4*HIGH_W*LOW_Z*LOW_Z +
     a7*b4*HIGH_W*HIGH_W*LOW_Z*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z +
a7*b1*LOW_W*MED_Z +
     a1*b2*LOW_W + a4*b2*LOW_W*LOW_W + a5*b2*MED_Z*LOW_W +
     a7*b2*LOW_W*LOW_W*MED_Z + a1*b3*MED_Z + a4*b3*LOW_W*MED_Z
+
     a5*b3*MED_Z*MED_Z + a7*b3*LOW_W*MED_Z*MED_Z +
a1*b4*LOW_W*MED_Z +
     a4*b4*LOW_W*LOW_W*MED_Z + a5*b4*LOW_W*MED_Z*MED_Z +
     a7*b4*LOW_W*LOW_W*MED_Z*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z +
a7*b1*MED_W*MED_Z +
     a1*b2*MED_W + a4*b2*MED_W*MED_W + a5*b2*MED_Z*MED_W +
     a7*b2*MED_W*MED_W*MED_Z + a1*b3*MED_Z + a4*b3*MED_W*MED_Z
+
     a5*b3*MED_Z*MED_Z + a7*b3*MED_W*MED_Z*MED_Z +
a1*b4*MED_W*MED_Z +
     a4*b4*MED_W*MED_W*MED_Z + a5*b4*MED_W*MED_Z*MED_Z +
     a7*b4*MED_W*MED_W*MED_Z*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a7*b1*HIGH_W*MED_Z +
     a1*b2*HIGH_W + a4*b2*HIGH_W*HIGH_W + a5*b2*MED_Z*HIGH_W
+
     a7*b2*HIGH_W*HIGH_W*MED_Z + a1*b3*MED_Z +
a4*b3*HIGH_W*MED_Z +
     a5*b3*MED_Z*MED_Z + a7*b3*HIGH_W*MED_Z*MED_Z +
a1*b4*HIGH_W*MED_Z +
     a4*b4*HIGH_W*HIGH_W*MED_Z + a5*b4*HIGH_W*MED_Z*MED_Z +
     a7*b4*HIGH_W*HIGH_W*MED_Z*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a7*b1*LOW_W*HIGH_Z +
     a1*b2*LOW_W + a4*b2*LOW_W*LOW_W + a5*b2*HIGH_Z*LOW_W +
     a7*b2*LOW_W*LOW_W*HIGH_Z + a1*b3*HIGH_Z +
a4*b3*LOW_W*HIGH_Z +
     a5*b3*HIGH_Z*HIGH_Z + a7*b3*LOW_W*HIGH_Z*HIGH_Z +
a1*b4*LOW_W*HIGH_Z +
     a4*b4*LOW_W*LOW_W*HIGH_Z + a5*b4*LOW_W*HIGH_Z*HIGH_Z +
     a7*b4*LOW_W*LOW_W*HIGH_Z*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a7*b1*MED_W*HIGH_Z +
     a1*b2*MED_W + a4*b2*MED_W*MED_W + a5*b2*HIGH_Z*MED_W +
     a7*b2*MED_W*MED_W*HIGH_Z + a1*b3*HIGH_Z +
a4*b3*MED_W*HIGH_Z +
     a5*b3*HIGH_Z*HIGH_Z + a7*b3*MED_W*HIGH_Z*HIGH_Z +
a1*b4*MED_W*HIGH_Z +
```

```
     a4*b4*MED_W*MED_W*HIGH_Z + a5*b4*MED_W*HIGH_Z*HIGH_Z +
     a7*b4*MED_W*MED_W*HIGH_Z*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a7*b1*HIGH_W*HIGH_Z +
     a1*b2*HIGH_W + a4*b2*HIGH_W*HIGH_W + a5*b2*HIGH_Z*HIGH_W
+
     a7*b2*HIGH_W*HIGH_W*HIGH_Z + a1*b3*HIGH_Z +
a4*b3*HIGH_W*HIGH_Z +
     a5*b3*HIGH_Z*HIGH_Z + a7*b3*HIGH_W*HIGH_Z*HIGH_Z +
a1*b4*HIGH_W*HIGH_Z +
     a4*b4*HIGH_W*HIGH_W*HIGH_Z + a5*b4*HIGH_W*HIGH_Z*HIGH_Z
+
     a7*b4*HIGH_W*HIGH_W*HIGH_Z*HIGH_Z;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z +
cdash7*LOW_W*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z +
cdash7*MED_W*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z +
cdash7*HIGH_W*LOW_Z;

    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z +
cdash7*LOW_W*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z +
cdash7*MED_W*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z +
cdash7*HIGH_W*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z +
cdash7*LOW_W*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z +
cdash7*MED_W*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z +
cdash7*HIGH_W*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + DLOW_LOZ;
    TMEW_LOZ = IMEW_LOZ + DMEW_LOZ;
    THIW_LOZ = IHIW_LOZ + DHIW_LOZ;

    TLOW_MEZ = ILOW_MEZ + DLOW_MEZ;
    TMEW_MEZ = IMEW_MEZ + DMEW_MEZ;
    THIW_MEZ = IHIW_MEZ + DHIW_MEZ;
```

```
    TLOW_HIZ = ILOW_HIZ + DLOW_HIZ;
    TMEW_HIZ = IMEW_HIZ + DMEW_HIZ;
    THIW_HIZ = IHIW_HIZ + DHIW_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 74: 1 or more mediators, in parallel if multiple (example uses 1), IV also moderates the Mediator-DV path

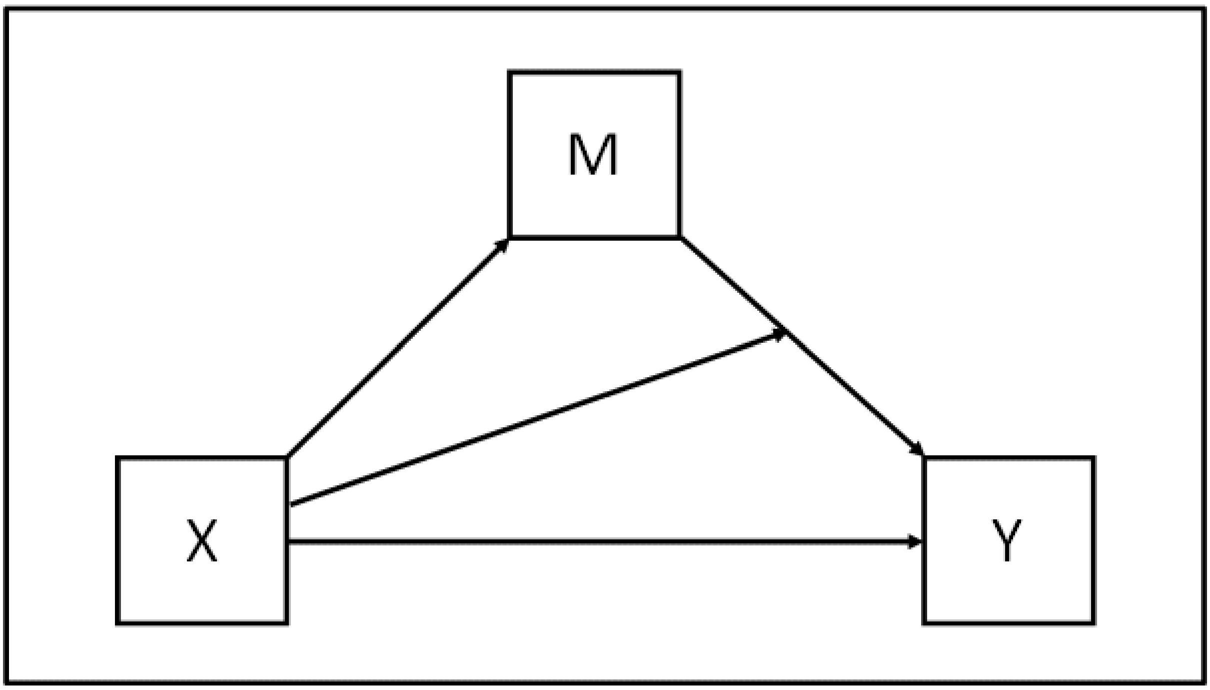
Example Variables: 1 predictor X, 1 mediator M, 1 outcome Y

Preliminary notes:

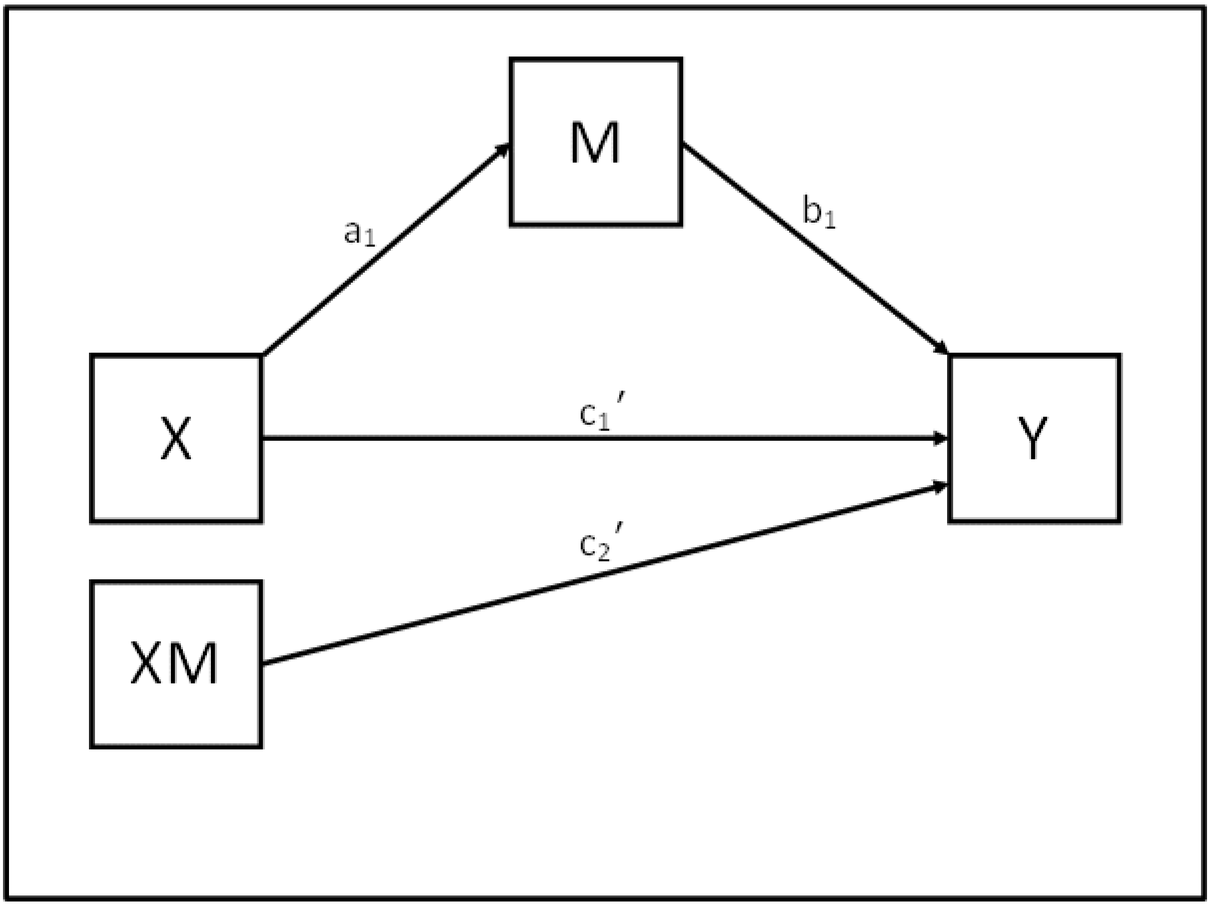
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- The mediator (variable M) is continuous. An example of how to handle a dichotomous mediator is given in model 4c.
- The DV (variable Y) is continuous and satisfies the assumptions of standard multiple regression. An example of how to handle a dichotomous DV is given in model 1e (i.e. a moderated logistic regression) and in model 4d (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c1'X + c2'MX
M = a0 + a1X

Algebra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c1'X + c2'MX
M = a0 + a1X

Hence... substituting in equations for M

Y = b0 + b1(a0 + a1X) + c1'X + c2'(a0 + a1X)X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + c1'X + a0c2'X + a1c2'XX

Hence... grouping terms into form Y = a + bX

Y = (b0 + a1b1) + (a1b1 + c1' + a0c2' + a1c2'X)X

Hence...

Conditional Indirect effect of X on Y:

a1b1 + a1c2'X = a1*(b1 + c2'X)

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) - (X)
! Outcome variable - Y

USEVARIABLES = X M Y XM;

DEFINE:
   XM = X*M;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path using parentheses

MODEL:
   Y ON M (b1);

   Y ON X (cdash1);    ! direct effect of X on Y
   Y ON XM (cdash2);

   [M] (a0);
   M ON X (a1);

! Use model constraint to calculate indirect effect

MODEL CONSTRAINT:
   NEW(LOW_X MED_X HIGH_X IND_LOWX IND_MEDX IND_HIX);

    LOW_X = #LOWX;    ! replace #LOWX in the code with your
chosen low value of X
    MED_X = #MEDX;    ! replace #MEDX in the code with your
chosen medium value of X
    HIGH_X = #HIGHX;    ! replace #HIGHX in the code with your
chosen high value of X
```

```
! Calc conditional indirect effects of X on Y via M for low,
medium, high values of X

    IND_LOWX = a1*b1 + a1*cdash2*LOW_X;
    IND_MEDX = a1*b1 + a1*cdash2*MED_X;
    IND_HIX = a1*b1 + a1*cdash2*HIGH_X;

! Use loop plot to plot conditional indirect effect of X on Y
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(INDX);

    LOOP(XVAL,1,5,0.1);

    INDX = (a1*b1 + a1*cdash2*XVAL)*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 75: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating both the IV- Mediator path and the Mediator-DV path

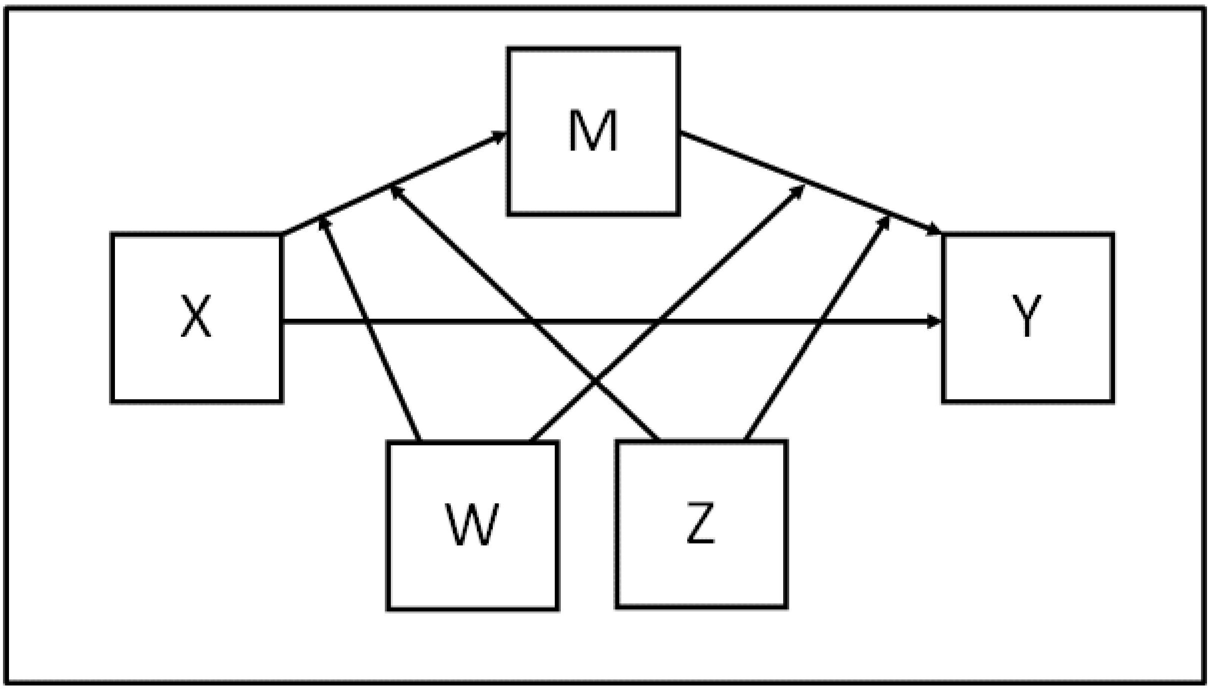
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

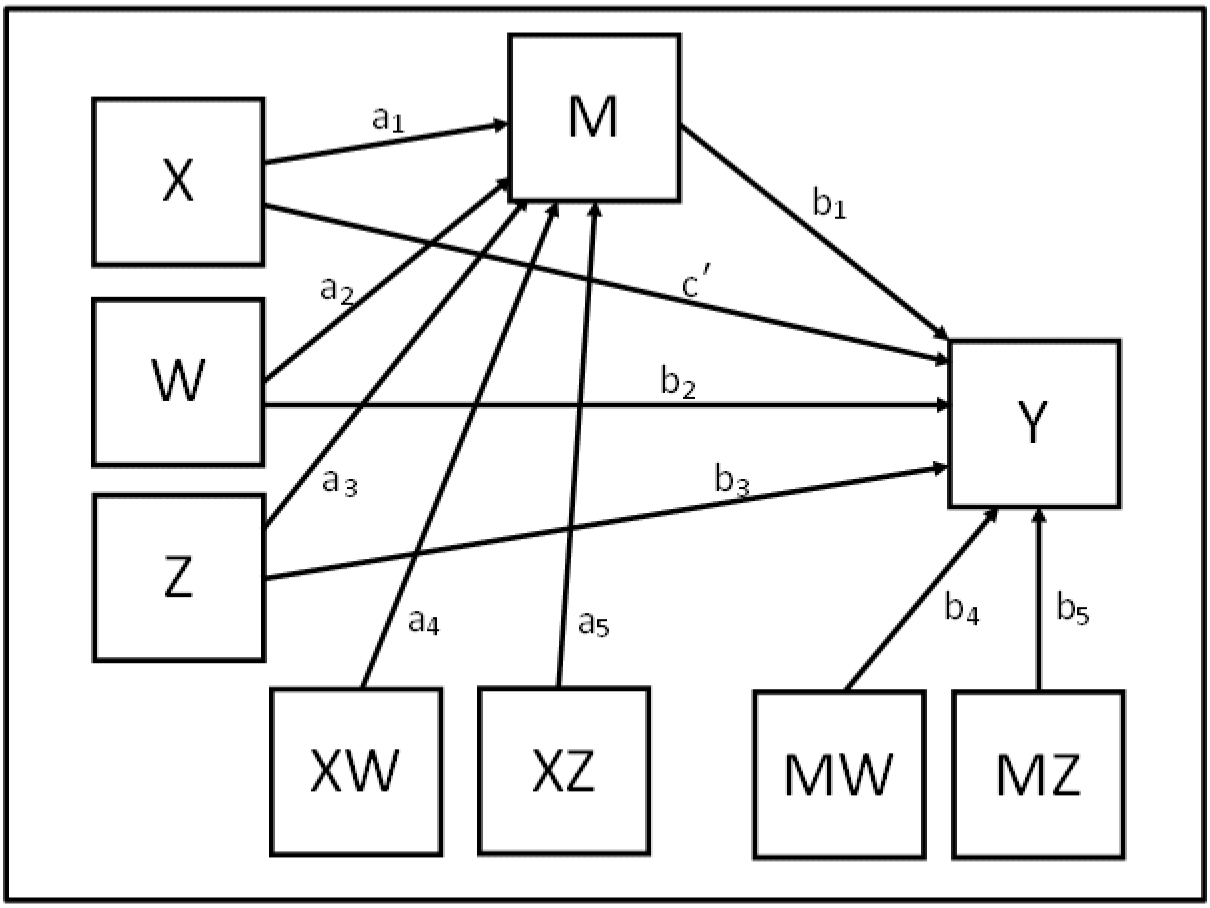
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2W + b3Z + b4MW + b5MZ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2W + b3Z + b4MW + b5MZ + c'X
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2W + b3Z + b4(a0 + a1X + a2W + a3Z + a4XW + a5XZ)W + b5(a0 + a1X + a2W + a3Z + a4XW + a5XZ)Z + c'X

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + b2W + b3Z + a0b4W + a1b4XW + a2b4WW + a3b4ZW + a4b4XWW + a5b4XZW + a0b5Z + a1b5XZ + a2b5WZ + a3b5ZZ + a4b5XWZ + a5b5XZZ + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + b2W + b3Z + a0b4W + a2b4WW + a3b4ZW + a0b5Z + a2b5WZ + a3b5ZZ) + (a1b1 + a4b1W + a5b1Z + a1b4W + a4b4WW + a5b4ZW + a1b5Z + a4b5WZ + a5b5ZZ + c')X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z:

a1b1 + a4b1W + a5b1Z + a1b4W + a4b4WW + a5b4ZW + a1b5Z + a4b5WZ + a5b5ZZ = (a1 + a4W + a5Z)(b1 + b4W + b5Z)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M
! Moderator variable(s) - W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ MW MZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MZ = M*Z;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
   Y ON W (b2);
   Y ON Z (b3);
```

```
    Y ON MW (b4);
    Y ON MZ (b5);

    Y ON X(cdash);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z

! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b4*LOW_W
+
     a4*b4*LOW_W*LOW_W + a5*b4*LOW_Z*LOW_W + a1*b5*LOW_Z +
     a4*b5*LOW_W*LOW_Z + a5*b5*LOW_Z*LOW_Z;
```

```
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b4*MED_W
+
     a4*b4*MED_W*MED_W + a5*b4*LOW_Z*MED_W + a1*b5*LOW_Z +
     a4*b5*MED_W*LOW_Z + a5*b5*LOW_Z*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a1*b4*HIGH_W +
     a4*b4*HIGH_W*HIGH_W + a5*b4*LOW_Z*HIGH_W + a1*b5*LOW_Z +
     a4*b5*HIGH_W*LOW_Z + a5*b5*LOW_Z*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b4*LOW_W
+
     a4*b4*LOW_W*LOW_W + a5*b4*MED_Z*LOW_W + a1*b5*MED_Z +
     a4*b5*LOW_W*MED_Z + a5*b5*MED_Z*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b4*MED_W
+
     a4*b4*MED_W*MED_W + a5*b4*MED_Z*MED_W + a1*b5*MED_Z +
     a4*b5*MED_W*MED_Z + a5*b5*MED_Z*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b4*HIGH_W +
     a4*b4*HIGH_W*HIGH_W + a5*b4*MED_Z*HIGH_W + a1*b5*MED_Z +
     a4*b5*HIGH_W*MED_Z + a5*b5*MED_Z*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b4*LOW_W +
     a4*b4*LOW_W*LOW_W + a5*b4*HIGH_Z*LOW_W + a1*b5*HIGH_Z +
     a4*b5*LOW_W*HIGH_Z + a5*b5*HIGH_Z*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b4*MED_W +
     a4*b4*MED_W*MED_W + a5*b4*HIGH_Z*MED_W + a1*b5*HIGH_Z +
     a4*b5*MED_W*HIGH_Z + a5*b5*HIGH_Z*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b4*HIGH_W +
     a4*b4*HIGH_W*HIGH_W + a5*b4*HIGH_Z*HIGH_W + a1*b5*HIGH_Z
+
     a4*b5*HIGH_W*HIGH_Z + a5*b5*HIGH_Z*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + cdash;
    TMEW_LOZ = IMEW_LOZ + cdash;
    THIW_LOZ = IHIW_LOZ + cdash;

    TLOW_MEZ = ILOW_MEZ + cdash;
    TMEW_MEZ = IMEW_MEZ + cdash;
    THIW_MEZ = IHIW_MEZ + cdash;

    TLOW_HIZ = ILOW_HIZ + cdash;
    TMEW_HIZ = IMEW_HIZ + cdash;
    THIW_HIZ = IHIW_HIZ + cdash;
```

```
! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 76: 1 or more mediators, in parallel if multiple (example uses 1), 2 moderators, both moderating each of the IV-Mediator path, the Mediator-DV path and the direct IV-DV path

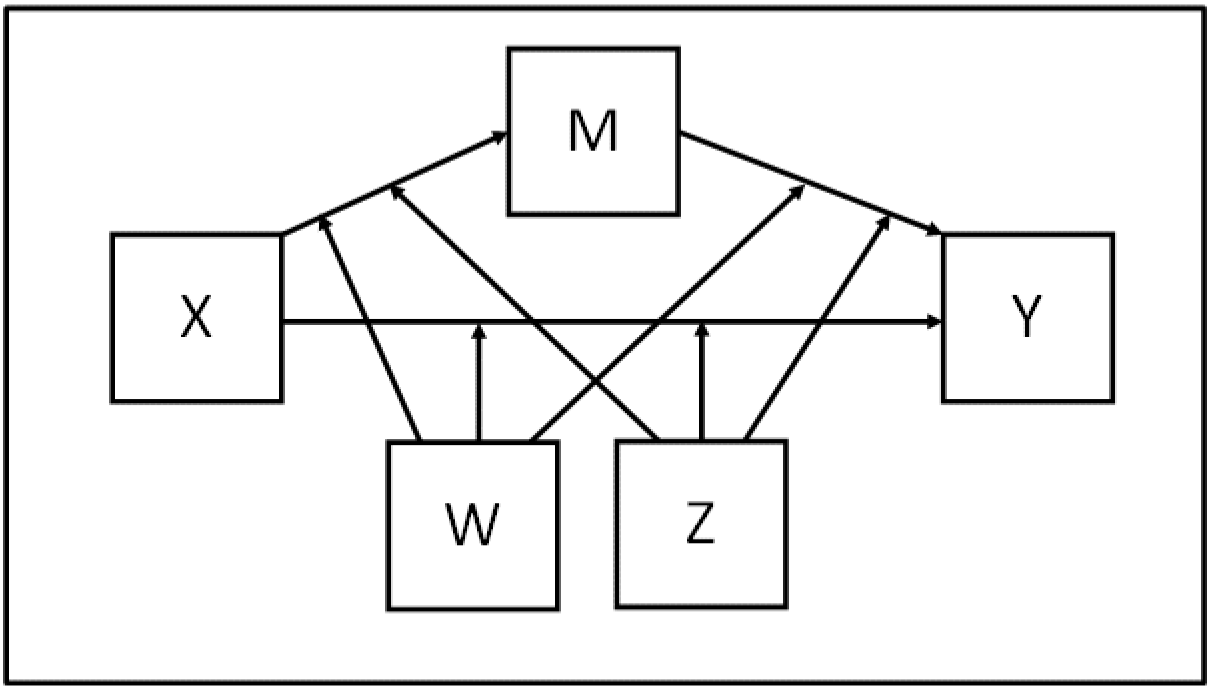
Example Variables: 1 predictor X, 1 mediator M, 2 moderators W, Z, 1 outcome Y

Preliminary notes:

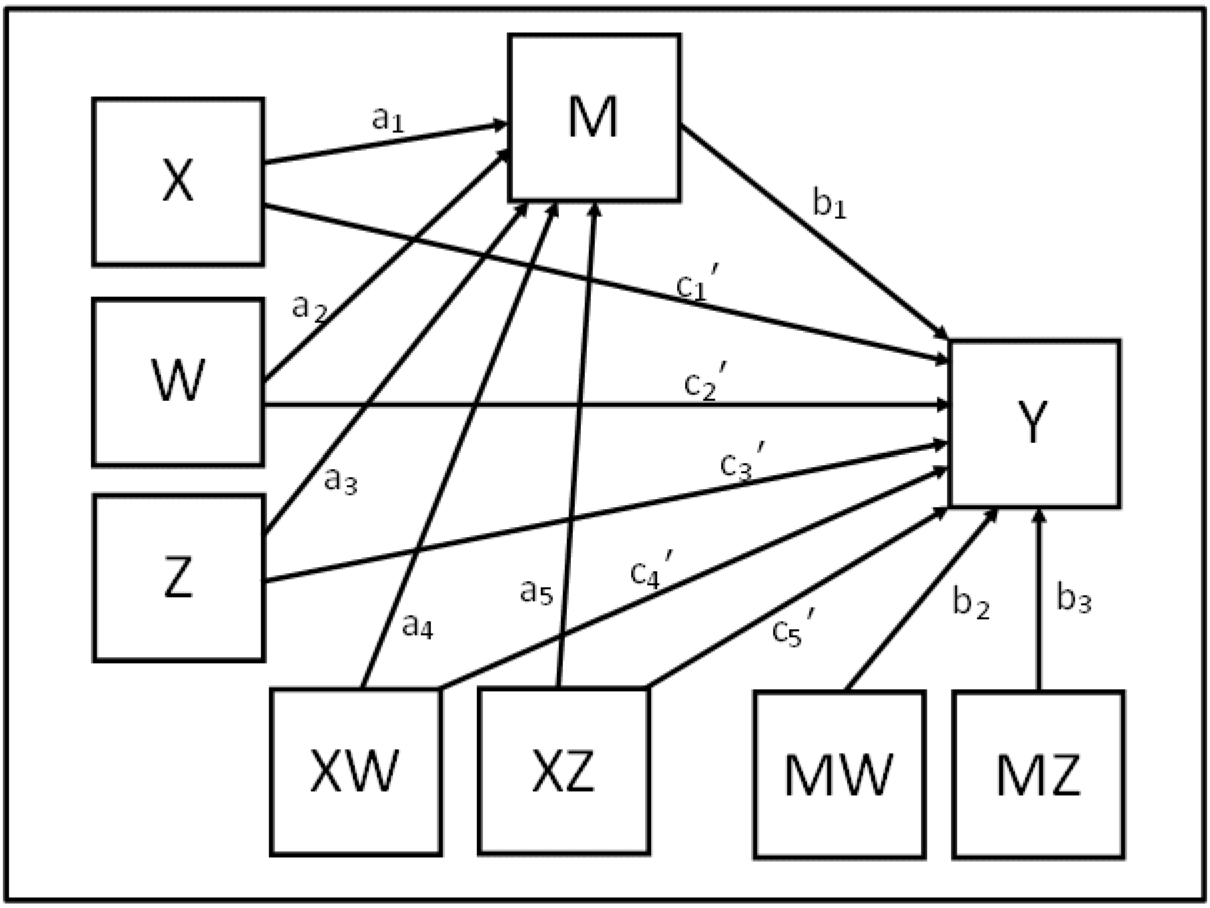
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous.
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation).
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous.
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + b2MW + b3MZ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Algebra to calculate indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + b2MW + b3MZ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ
M = a0 + a1X + a2W + a3Z + a4XW + a5XZ

Hence... substituting in equation for M

Y = b0 + b1(a0 + a1X + a2W + a3Z + a4XW + a5XZ) + b2(a0 + a1X + a2W + a3Z + a4XW + a5XZ)W + b3(a0 + a1X + a2W + a3Z + a4XW + a5XZ)Z + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... multiplying out brackets

Y = b0 + a0b1 + a1b1X + a2b1W + a3b1Z + a4b1XW + a5b1XZ + a0b2W + a1b2XW + a2b2WW + a3b2ZW + a4b2XWW + a5b2XZW + a0b3Z + a1b3XZ + a2b3WZ + a3b3ZZ + a4b3XWZ + a5b3XZZ + c1'X + c2'W + c3'Z + c4'XW + c5'XZ

Hence... grouping terms into form Y = a + bX

Y = (b0 + a0b1 + a2b1W + a3b1Z + a0b2W + a2b2WW + a3b2ZW + a0b3Z + a2b3WZ + a3b3ZZ + c2'W + c3'Z) + (a1b1 + a4b1W + a5b1Z + a1b2W + a4b2WW + a5b2ZW + a1b3Z + a4b3WZ + a5b3ZZ + c1' + c4'W + c5'Z)X

Hence...

One indirect effect(s) of X on Y, conditional on W, Z: a1b1 + a4b1W + a5b1Z + a1b2W + a4b2WW + a5b2ZW + a1b3Z + a4b3WZ + a5b3ZZ = (a1 + a4W + a5Z)(b1 + b2W + b3Z)

One direct effect of X on Y, conditional on W, Z: c1' + c4'W + c5'Z

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M
! Moderator variable(s) – W, Z
! Outcome variable - Y

USEVARIABLES = X M W Z Y XW XZ MW MZ;

! Create interaction terms
! Note that they have to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   MW = M*W;
   MZ = M*Z;
   XW = X*W;
   XZ = X*Z;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path and intercept using
parentheses

MODEL:
   [Y] (b0);
   Y ON M (b1);
```

```
    Y ON MW (b2);
    Y ON MZ (b3);

    Y ON X(cdash1);
    Y ON W (cdash2);
    Y ON Z (cdash3);
    Y ON XW (cdash4);
    Y ON XZ (cdash5);

    [M] (a0);
    M ON X (a1);
    M ON W (a2);
    M ON Z (a3);
    M ON XW (a4);
    M ON XZ (a5);

! Use model constraint subcommand to test conditional indirect
effects
! You need to pick low, medium and high moderator values for W,
Z
! for example, of 1 SD below mean, mean, 1 SD above mean

! 2 moderators, 3 values for each, gives 9 combinations
! arbitrary naming convention for conditional indirect and
total effects used below:
! MEV_LOQ = medium value of V and low value of Q, etc.

MODEL CONSTRAINT:
    NEW(LOW_W MED_W HIGH_W LOW_Z MED_Z HIGH_Z
    ILOW_LOZ IMEW_LOZ IHIW_LOZ ILOW_MEZ IMEW_MEZ IHIW_MEZ
    ILOW_HIZ IMEW_HIZ IHIW_HIZ
    DLOW_LOZ DMEW_LOZ DHIW_LOZ DLOW_MEZ DMEW_MEZ DHIW_MEZ
    DLOW_HIZ DMEW_HIZ DHIW_HIZ
    TLOW_LOZ TMEW_LOZ THIW_LOZ TLOW_MEZ TMEW_MEZ THIW_MEZ
    TLOW_HIZ TMEW_HIZ THIW_HIZ);

    LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
    MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
    HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

    LOW_Z = #LOWZ;   ! replace #LOWZ in the code with your
chosen low value of Z
    MED_Z = #MEDZ;   ! replace #MEDZ in the code with your
chosen medium value of Z
    HIGH_Z = #HIGHZ;   ! replace #HIGHZ in the code with your
chosen high value of Z
```

```
! Calc conditional indirect effects for each combination of
moderator values

    ILOW_LOZ = a1*b1 + a4*b1*LOW_W + a5*b1*LOW_Z + a1*b2*LOW_W
+
     a4*b2*LOW_W*LOW_W + a5*b2*LOW_Z*LOW_W + a1*b3*LOW_Z +
     a4*b3*LOW_W*LOW_Z + a5*b3*LOW_Z*LOW_Z;
    IMEW_LOZ = a1*b1 + a4*b1*MED_W + a5*b1*LOW_Z + a1*b2*MED_W
+
     a4*b2*MED_W*MED_W + a5*b2*LOW_Z*MED_W + a1*b3*LOW_Z +
     a4*b3*MED_W*LOW_Z + a5*b3*LOW_Z*LOW_Z;
    IHIW_LOZ = a1*b1 + a4*b1*HIGH_W + a5*b1*LOW_Z +
a1*b2*HIGH_W +
     a4*b2*HIGH_W*HIGH_W + a5*b2*LOW_Z*HIGH_W + a1*b3*LOW_Z +
     a4*b3*HIGH_W*LOW_Z + a5*b3*LOW_Z*LOW_Z;

    ILOW_MEZ = a1*b1 + a4*b1*LOW_W + a5*b1*MED_Z + a1*b2*LOW_W
+
     a4*b2*LOW_W*LOW_W + a5*b2*MED_Z*LOW_W + a1*b3*MED_Z +
     a4*b3*LOW_W*MED_Z + a5*b3*MED_Z*MED_Z;
    IMEW_MEZ = a1*b1 + a4*b1*MED_W + a5*b1*MED_Z + a1*b2*MED_W
+
     a4*b2*MED_W*MED_W + a5*b2*MED_Z*MED_W + a1*b3*MED_Z +
     a4*b3*MED_W*MED_Z + a5*b3*MED_Z*MED_Z;
    IHIW_MEZ = a1*b1 + a4*b1*HIGH_W + a5*b1*MED_Z +
a1*b2*HIGH_W +
     a4*b2*HIGH_W*HIGH_W + a5*b2*MED_Z*HIGH_W + a1*b3*MED_Z +
     a4*b3*HIGH_W*MED_Z + a5*b3*MED_Z*MED_Z;

    ILOW_HIZ = a1*b1 + a4*b1*LOW_W + a5*b1*HIGH_Z +
a1*b2*LOW_W +
     a4*b2*LOW_W*LOW_W + a5*b2*HIGH_Z*LOW_W + a1*b3*HIGH_Z +
     a4*b3*LOW_W*HIGH_Z + a5*b3*HIGH_Z*HIGH_Z;
    IMEW_HIZ = a1*b1 + a4*b1*MED_W + a5*b1*HIGH_Z +
a1*b2*MED_W +
     a4*b2*MED_W*MED_W + a5*b2*HIGH_Z*MED_W + a1*b3*HIGH_Z +
     a4*b3*MED_W*HIGH_Z + a5*b3*HIGH_Z*HIGH_Z;
    IHIW_HIZ = a1*b1 + a4*b1*HIGH_W + a5*b1*HIGH_Z +
a1*b2*HIGH_W +
     a4*b2*HIGH_W*HIGH_W + a5*b2*HIGH_Z*HIGH_W + a1*b3*HIGH_Z
+
     a4*b3*HIGH_W*HIGH_Z + a5*b3*HIGH_Z*HIGH_Z;

! Calc conditional direct effects for each combination of
moderator values

    DLOW_LOZ = cdash1 + cdash4*LOW_W + cdash5*LOW_Z;
    DMEW_LOZ = cdash1 + cdash4*MED_W + cdash5*LOW_Z;
    DHIW_LOZ = cdash1 + cdash4*HIGH_W + cdash5*LOW_Z;
```

```
    DLOW_MEZ = cdash1 + cdash4*LOW_W + cdash5*MED_Z;
    DMEW_MEZ = cdash1 + cdash4*MED_W + cdash5*MED_Z;
    DHIW_MEZ = cdash1 + cdash4*HIGH_W + cdash5*MED_Z;

    DLOW_HIZ = cdash1 + cdash4*LOW_W + cdash5*HIGH_Z;
    DMEW_HIZ = cdash1 + cdash4*MED_W + cdash5*HIGH_Z;
    DHIW_HIZ = cdash1 + cdash4*HIGH_W + cdash5*HIGH_Z;

! Calc conditional total effects for each combination of
moderator values

    TLOW_LOZ = ILOW_LOZ + DLOW_LOZ;
    TMEW_LOZ = IMEW_LOZ + DMEW_LOZ;
    THIW_LOZ = IHIW_LOZ + DHIW_LOZ;

    TLOW_MEZ = ILOW_MEZ + DLOW_MEZ;
    TMEW_MEZ = IMEW_MEZ + DMEW_MEZ;
    THIW_MEZ = IHIW_MEZ + DHIW_MEZ;

    TLOW_HIZ = ILOW_HIZ + DLOW_HIZ;
    TMEW_HIZ = IMEW_HIZ + DMEW_HIZ;
    THIW_HIZ = IHIW_HIZ + DHIW_HIZ;

! Use loop plot to plot conditional indirect effect of X on Y
for each combination of low, med, high moderator values
! Could be edited to show conditional direct or conditional
total effects instead
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(PLOW_LOZ PMEW_LOZ PHIW_LOZ PLOW_MEZ PMEW_MEZ
PHIW_MEZ
    PLOW_HIZ PMEW_HIZ PHIW_HIZ);

    LOOP(XVAL,1,5,0.1);

    PLOW_LOZ = ILOW_LOZ*XVAL;
    PMEW_LOZ = IMEW_LOZ*XVAL;
    PHIW_LOZ = IHIW_LOZ*XVAL;

    PLOW_MEZ = ILOW_MEZ*XVAL;
    PMEW_MEZ = IMEW_MEZ*XVAL;
    PHIW_MEZ = IHIW_MEZ*XVAL;

    PLOW_HIZ = ILOW_HIZ*XVAL;
    PMEW_HIZ = IMEW_HIZ*XVAL;
    PHIW_HIZ = IHIW_HIZ*XVAL;

PLOT:
   TYPE = plot2;
```

```
OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 77: 3 or more mediators, both in parallel and in series

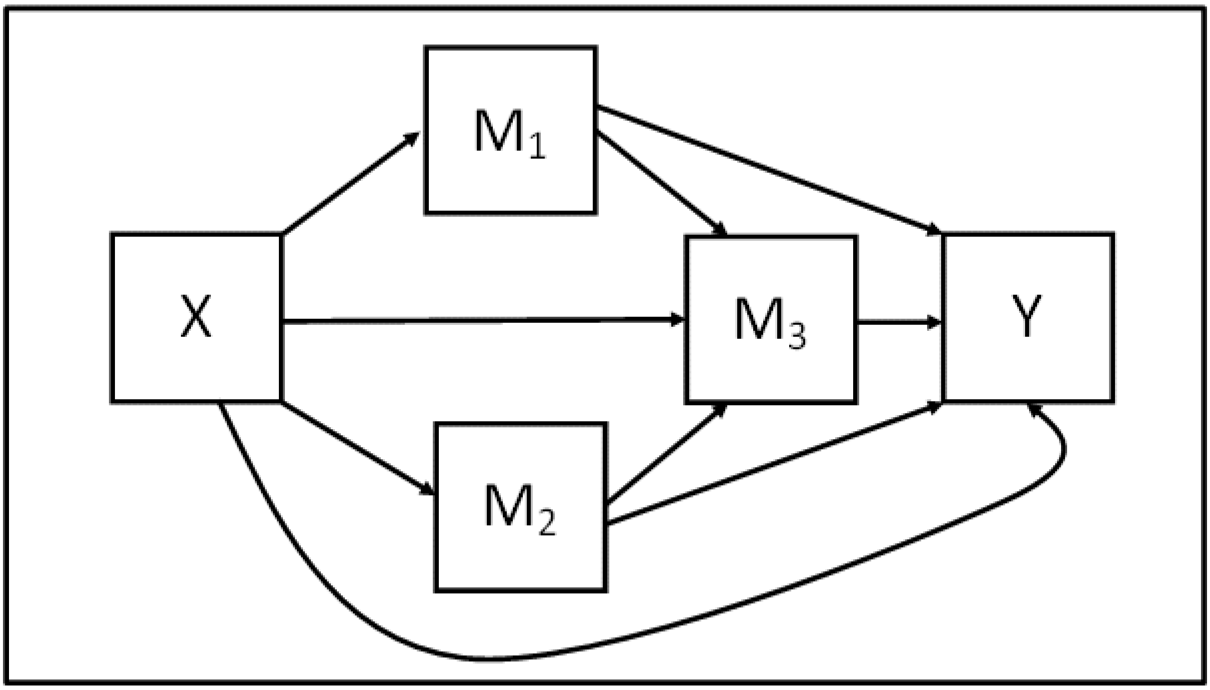
Example Variables: 1 predictor X, 3 mediators M1, M2, and M3, 1 outcome Y

Preliminary notes:

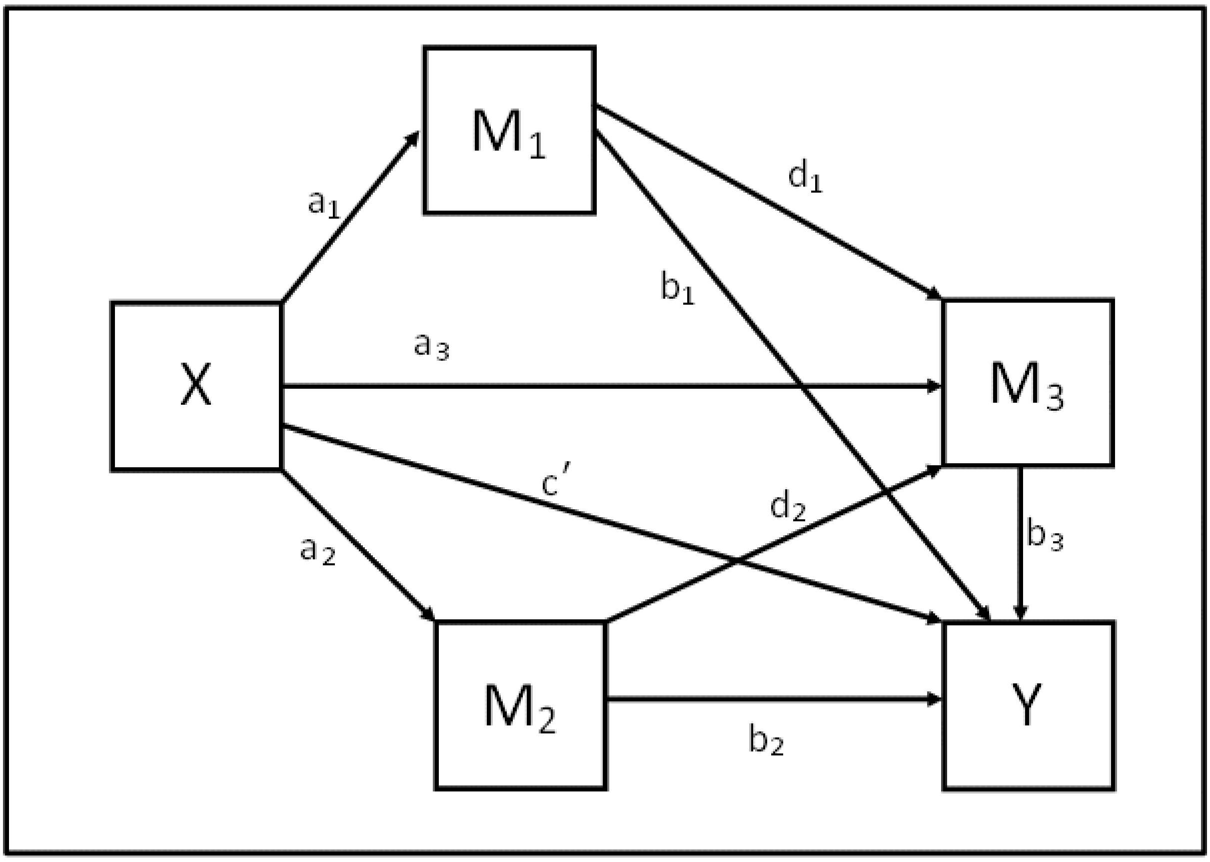
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- Any moderators (variables W,V,Q,Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation)
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M1 + b2M2 + b3M3 + c'X
M1 = a01 + a1X
M2 = a02 + a2X
M3 = a03 + a3X + d1M1 + d2M2

Albegra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M1 + b2M2 + b3M3 + c'X
M1 = a01 + a1X
M2 = a02 + a2X
M3 = a03 + a3X + d1M1 + d2M2

Hence... substituting in equations for M1 and M2 into Y and M3

Y = b0 + b1(a01 + a1X) + b2(a02 + a2X) + b3M3 + c'X
M3 = a03 + a3X + d1(a01 + a1X) + d2(a02 + a2X)

Hence... substituting in equations for M3 into Y

Y = b0 + b1(a01 + a1X) + b2(a02 + a2X) + b3(a03 + a3X + d1(a01 + a1X) + d2(a02 + a2X)) + c'X

Hence... multiplying out brackets

Y = b0 + a01b1 + a1b1X + a02b2 + a2b2X + a03b3 + a3b3X + a01b3d1 + a1b3d1X + a02b3d2 + a2b3d2X + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a01b1 + a02b2 + a03b3 + a01b3d1 + a02b3d2) + (a1b1 + a2b2 + a3b3 + a1b3d1 + a2b3d2 + c')X

Hence...

Five indirect effects of X on Y:

a1b1, a2b2, a3b3, a1b3d1, a2b3d2

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M1, M2, M3
! Moderator variable(s) - none
! Outcome variable - Y

USEVARIABLES = X M1 M2 Y;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path using parentheses

MODEL:
   Y ON M1 (b1);
   Y ON M2 (b2);
   Y ON M3 (b3);

   Y ON X (cdash);    ! direct effect of X on Y

   M1 ON X (a1);
   M2 ON X (a2);
   M3 ON X (a3);
```

```
   M3 ON M1 (d1);
   M3 ON M2 (d2);

! Use model constraint to calculate specific indirect paths
and total indirect effect

MODEL CONSTRAINT:
   NEW(a1b1 a2b2 a3b3 a1d1b3 a2d2b3 TOTALIND TOTAL);
   a1b1 = a1*b1;   ! Specific indirect effect of X on Y via M1
only
   a2b2 = a2*b2;   ! Specific indirect effect of X on Y via M2
only
   a3b3 = a3*b3;   ! Specific indirect effect of X on Y via M3
only
   a1d1b3 = a1*d1*b3;   ! Specific indirect effect of X on Y
via M1 and M3
   a2d2b3 = a2*d2*b3;   ! Specific indirect effect of X on Y
via M2 and M3
   TOTALIND = a1b1 + a2b2 + a3b3 + a1d1b3 + a2d2b3;   ! Total
indirect effect of X on Y via M1, M2, M3
   TOTAL = a1b1 + a2b2 + a3b3 + a1d1b3 + a2d2b3 + cdash;   !
Total effect of X on Y

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 78: 2 or more mediators, in series, 1 moderator moderating path between mediators

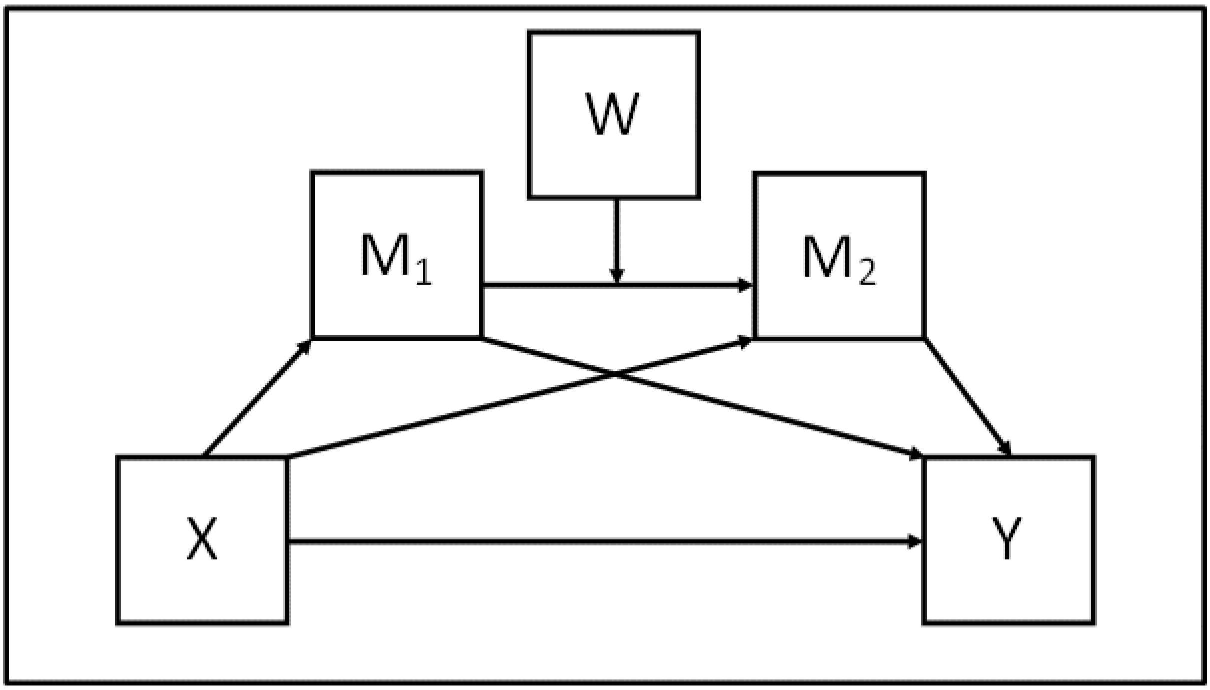
Example Variables: 1 predictor X, 2 mediators M1 and M2, 1 moderator W, 1 outcome Y

Preliminary notes:

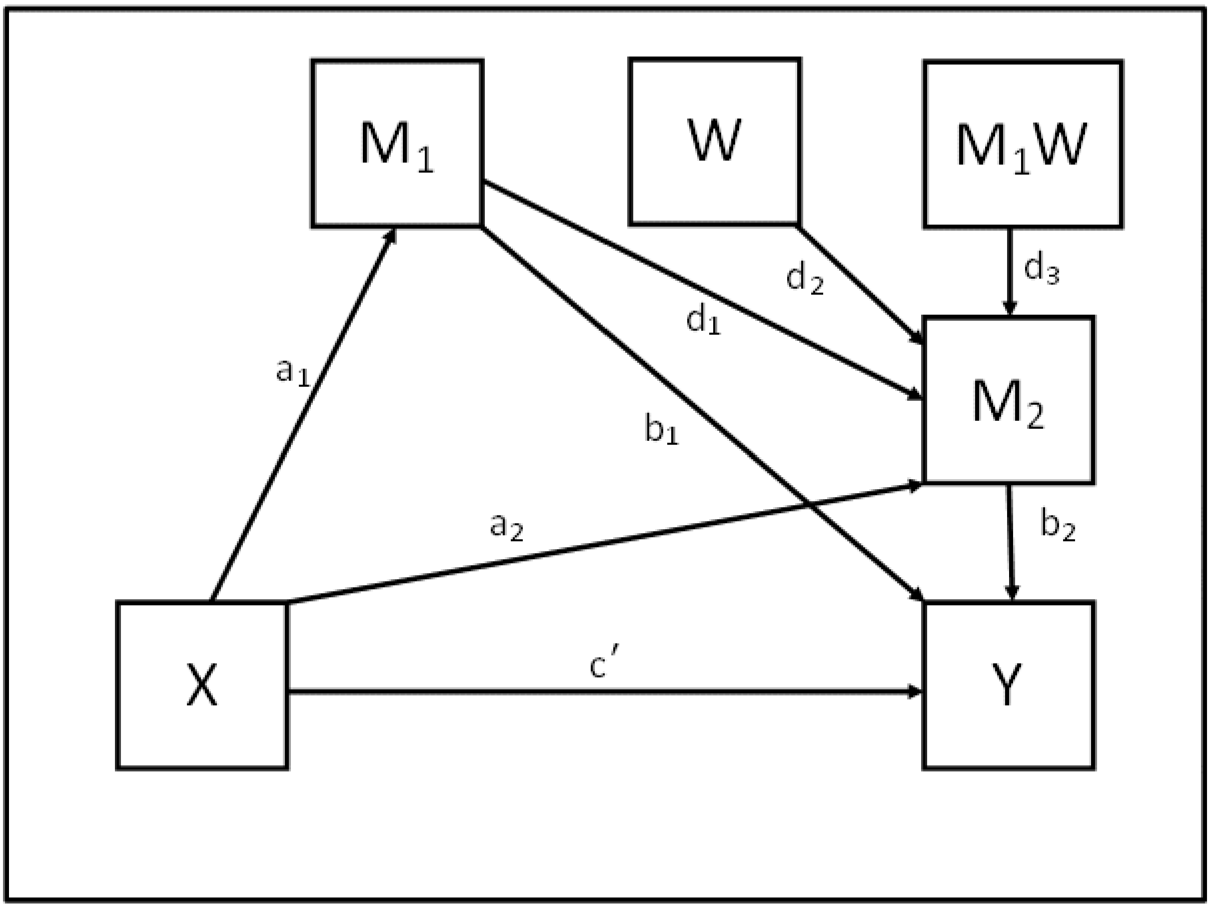
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- Any moderators (variables W,V,Q,Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation)
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M1 + b2M2 + c'X
M1 = a01 + a1X
M2 = a02 + a2X + d1M1 + d2W + d3M1W

Albegra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M1 + b2M2 + c'X
M1 = a01 + a1X
M2 = a02 + a2X + d1M1 + d2W + d3M1W

Hence... substituting in equations for M1 and M2

Y = b0 + b1(a01 + a1X) + b2(a02 + a2X + d1(a01 + a1X) + d2W + d3(a01 + a1X)W) + c'X

Hence... multiplying out brackets

Y = b0 + a01b1 + a1b1X + a02b2 + a2b2X + a01b2d1 + a1b2d1X + d2b2W + a01b2d3W + a1b2d3XW + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a01b1 + a02b2 + a01b2d1 + d2b2W + a01b2d3W) + (a1b1 + a1b2d1 + a2b2 + a1b2d3W + c')X

Hence...

Three indirect effects of X on Y:

a1b1, a2b2, a1b2(d1 + d3W)

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) – M1, M2
! Moderator variable(s) - W
! Outcome variable - Y

USEVARIABLES = X M1 M2 W Y M1W;

! Create interaction term
! Note that it has to be placed at end of USEVARIABLES
subcommand above

DEFINE:
   M1W = M1*W;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path using parentheses

MODEL:
   Y ON M1 (b1);
   Y ON M2 (b2);

   Y ON X (cdash);   ! direct effect of X on Y

   M1 ON X (a1);
   M2 ON X (a2);
   M2 ON M1 (d1);
```

```
   M2 ON W (d2);
   M2 ON M1W (d3);

! Use model constraint subcommand to test simple slopes
! You need to pick low, medium and high moderator values,
! for example, of 1 SD below mean, mean, 1 SD above mean
! Also calc total effects at lo, med, hi values of moderator

MODEL CONSTRAINT:
   NEW(LOW_W MED_W HIGH_W a1b1 a2b2 LWa1d1b2 MWa1d1b2
HWa1d1b2
   TOT_LOWW TOT_MEDW TOT_HIW);

   LOW_W = #LOWW;   ! replace #LOWW in the code with your
chosen low value of W
   MED_W = #MEDW;   ! replace #MEDW in the code with your
chosen medium value of W
   HIGH_W = #HIGHW;   ! replace #HIGHW in the code with your
chosen high value of W

! Now calc indirect and total effects for each value of W
   a1b1 = a1*b1;   ! Specific indirect effect of X on Y via M1
only
   a2b2 = a2*b2;   ! Specific indirect effect of X on Y via M2
only

! Conditional indirect effects of X on Y via M1 and M2 given
values of W

   LWa1d1b2 = a1*d1*b2 + a1*d3*b2*LOW_W;
   MWa1d1b2 = a1*d1*b2 + a1*d3*b2*MED_W;
   HWa1d1b2 = a1*d1*b2 + a1*d3*b2*HIGH_W;

! Conditional total effects of X on Y given values of W

   TOT_LOWW = LWa1d1b2 + a1b1 + a2b2 + cdash;
   TOT_MEDW = MWa1d1b2 + a1b1 + a2b2 + cdash ;
   TOT_HIW = HWa1d1b2 + a1b1 + a2b2 + cdash;

! Use loop plot to plot total effect of X on Y for low, med,
high values of W
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

   PLOT(LOMOD MEDMOD HIMOD);

   LOOP(XVAL,1,5,0.1);

   LOMOD = TOT_LOWW*XVAL;
   MEDMOD = TOT_MEDW*XVAL;
   HIMOD = TOT_HIW*XVAL;
```

```
PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 79: 3 or more mediators, both in parallel and in series, 2 moderators, 1 moderating paths between predictor and mediator, the second moderating paths between mediators, and between mediator and DV

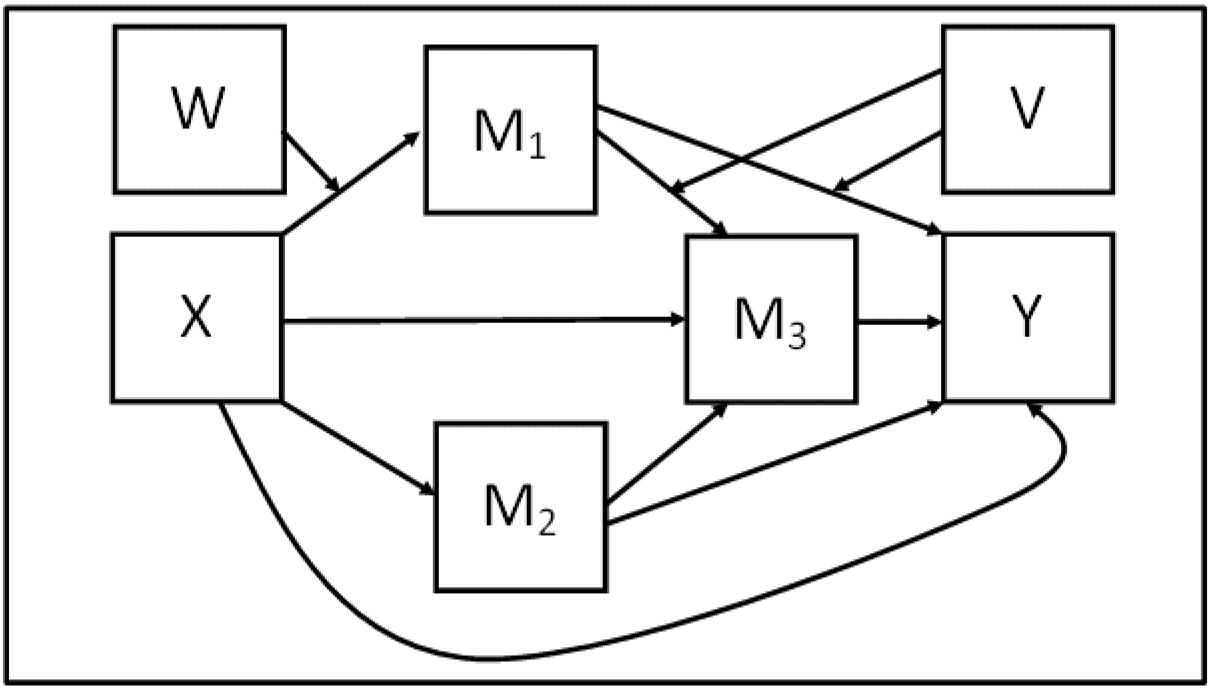
Example Variables: 1 predictor X, 3 mediators M1, M2, and M3, 2 moderators W, V, 1 outcome Y

Preliminary notes:

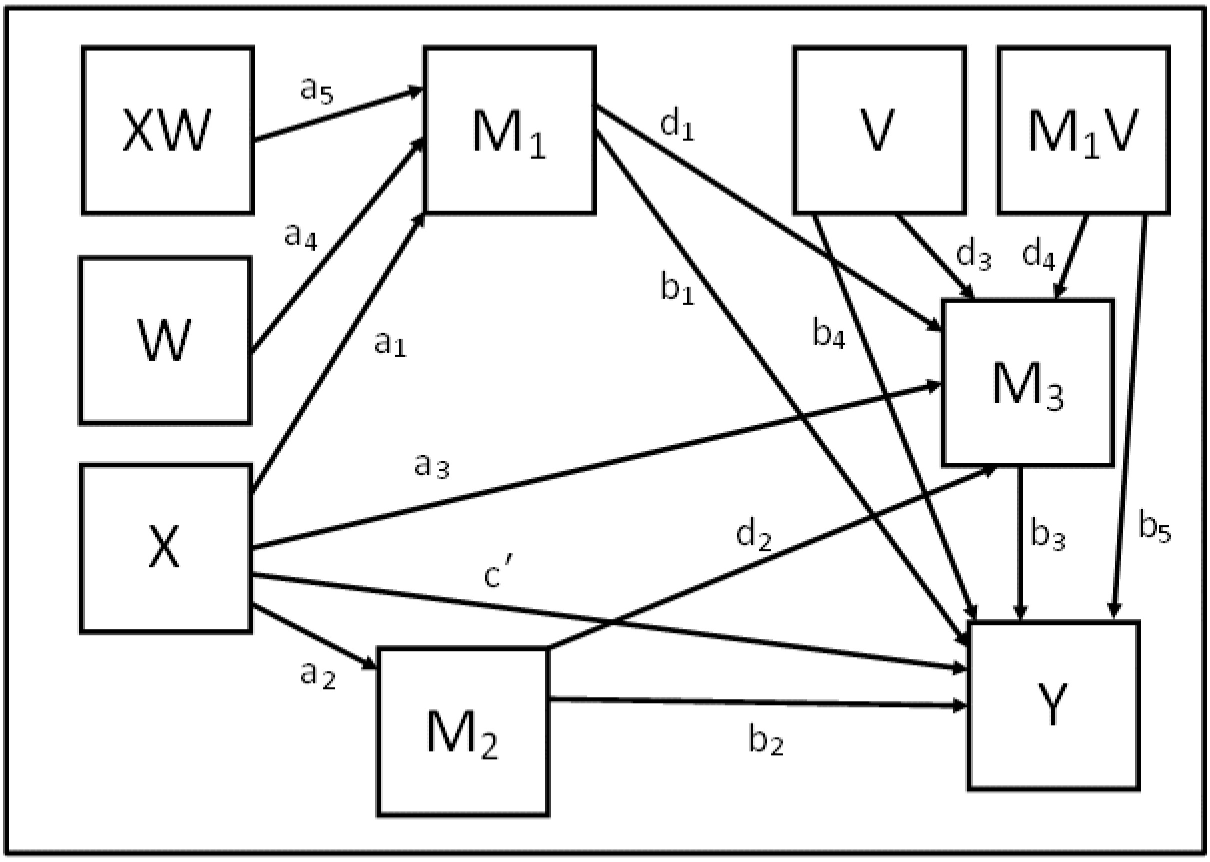
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- Any moderators (variables W,V,Q,Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given for model 1 (simple moderation)
- Any mediators (variable M, or M1, M2, etc.) are assumed to be continuous
- The DV (variable Y) is assumed to be continuous and to satisfy the assumptions of standard multiple regression - an example of how to handle a dichotomous DV is given for model 1 (i.e. a moderated logistic regression) and for model 4 (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M1 + b2M2 + b3M3 + b4V + b5M1V + c'X
M1 = a01 + a1X + a4W + a5XW
M2 = a02 + a2X
M3 = a03 + a3X + d1M1 + d2M2 + d3V + d4M1V

Albegra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M1 + b2M2 + b3M3 + b4V + b5M1V + c'X
M1 = a01 + a1X + a4W + a5XW
M2 = a02 + a2X
M3 = a03 + a3X + d1M1 + d2M2 + d3V + d4M1V

Hence... substituting in equations for M1 and M2 into Y and M3

Y = Y = b0 + b1(a01 + a1X + a4W + a5XW) + b2(a02 + a2X) + b3M3 + b4V + b5(a01 + a1X + a4W + a5XW)V + c'X
M3 = a03 + a3X + d1(a01 + a1X + a4W + a5XW) + d2(a02 + a2X) + d3V + d4(a01 + a1X + a4W + a5XW)V

Hence... substituting in equations for M3 into Y

Y = b0 + b1(a01 + a1X + a4W + a5XW) + b2(a02 + a2X) + b3(a03 + a3X + d1(a01 + a1X + a4W + a5XW) + d2(a02 + a2X) + d3V + d4(a01 + a1X + a4W + a5XW)V) + b4V + b5(a01 + a1X + a4W + a5XW)V + c'X

Hence... multiplying out brackets

Y = b0 + a01b1 + a1b1X + a4b1W + a5b1XW + a02b2 + a2b2X + a03b3 + a3b3X + a01d1b3 + a1d1b3X + a4d1b3W + a5d1b3XW + a02d2b3 + a2d2b3X + b3d3V + a01d4b3V + a1d4b3XV + a4d4b3WV + a5d4b3XWV + b4V + a01b5V + a1b5XV + a4b5WV + a5b5XWV + c'X

Hence... grouping terms into form Y = a + bX

Y = (b0 + a01b1 + a4b1W + a02b2 + a03b3 + a01d1b3 + a4d1b3W + a02d2b3 + b3d3V + a01d4b3V + a4d4b3WV + b4V + a01b5V + a4b5WV) + (a1b1 + a5b1W + a2b2 + a3b3 + a1d1b3 + a5d1b3W + a2d2b3 + a1d4b3V + a5d4b3WV + a1b5V + a5b5WV + c')X

Hence...

Five indirect effects of X on Y:

a1b1 + a5b1W + a1b5V + a5b5WV, a2b2, a3b3, a2d2b3, a1b3d1 + a5d1b3W + a1d4b3V + a5d4b3WV

One direct effect of X on Y:

c'

Mplus code for the model:

```
! Predictor variable - X
! Mediator variable(s) - M1, M2, M3
! Moderator variable(s) - W, V
! Outcome variable - Y

USEVARIABLES = X M1 M2 M3 W V Y XW M1V;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path using parentheses
```

```
MODEL:
   Y ON M1 (b1);
   Y ON M2 (b2);
   Y ON M3 (b3);
   Y ON V (b4);
   Y ON M1V (b5);

   Y ON X (cdash);    ! direct effect of X on Y

   M1 ON X (a1);
   M1 ON W (a4);
   M1 ON XW (a5);
   M2 ON X (a2);
   M3 ON X (a3);

   M3 ON M1 (d1);
   M3 ON M2 (d2);
   M3 ON V (d3);
   M3 ON M1V (d4);

! Use model constraint subcommand to test simple slopes
! You need to pick low, medium and high moderator values,
! for example, of 1 SD below mean, mean, 1 SD above mean
! Also calc total effects at lo, med, hi values of moderator

MODEL CONSTRAINT:
   NEW(LOW_W MED_W HIGH_W LOW_V MED_V HIGH_V
   a1b1LWLV a1b1MWLV a1b1HWLV a1b1LWMV a1b1MWMV a1b1HWMV
   a1b1LWHV a1b1MWHV a1b1HWHV
   a2b2 a3b3 a2d2b3
   adbLWLV adbMWLV adbHWLV adbLWMV adbMWMV adbHWMV
   adbLWHV adbMWHV adbHWHV
   TI_LWLV TI_MWLV TI_HWLV TI_LWMV TI_MWMV TI_HWMV
   TI_LWHV TI_MWHV TI_HWHV
   TOT_LWLV TOT_MWLV TOT_HWLV TOT_LWMV TOT_MWMV TOT_HWMV
   TOT_LWHV TOT_MWHV TOT_HWHV);

   LOW_W = #LOWW;    ! replace #LOWW in the code with your
chosen low value of W
   MED_W = #MEDW;    ! replace #MEDW in the code with your
chosen medium value of W
   HIGH_W = #HIGHW;    ! replace #HIGHW in the code with your
chosen high value of W

   LOW_V = #LOWV;    ! replace #LOWV in the code with your
chosen low value of V
   MED_V = #MEDV;    ! replace #MEDV in the code with your
chosen medium value of V
   HIGH_V = #HIGHV;    ! replace #HIGHV in the code with your
chosen high value of V
```

```
! Now calc specific indirect effects for each value of W and
V
   a1b1LWLV = a1*b1 + a5*b1*LOW_W + a1*b5*LOW_V +
a5*b5*LOW_W*LOW_V;
   a1b1MWLV = a1*b1 + a5*b1*MED_W + a1*b5*LOW_V +
a5*b5*MED_W*LOW_V;
   a1b1HWLV = a1*b1 + a5*b1*HIGH_W + a1*b5*LOW_V +
a5*b5*HIGH_W*LOW_V;
   a1b1LWMV = a1*b1 + a5*b1*LOW_W + a1*b5*MED_V +
a5*b5*LOW_W*MED_V;
   a1b1MWMV = a1*b1 + a5*b1*MED_W + a1*b5*MED_V +
a5*b5*MED_W*MED_V;
   a1b1HWMV = a1*b1 + a5*b1*HIGH_W + a1*b5*MED_V +
a5*b5*HIGH_W*MED_V;
   a1b1LWHV = a1*b1 + a5*b1*LOW_W + a1*b5*HIGH_V +
a5*b5*LOW_W*HIGH_V;
   a1b1MWHV = a1*b1 + a5*b1*MED_W + a1*b5*HIGH_V +
a5*b5*MED_W*HIGH_V;
   a1b1HWHV = a1*b1 + a5*b1*HIGH_W + a1*b5*HIGH_V +
a5*b5*HIGH_W*HIGH_V;

   a2b2 = a2*b2;

   a3b3 = a3*b3;

   a2d2b3 = a2*d2*b3;

   adbLWLV = a1*b3*d1 + a5*d1*b3*LOW_W + a1*d4*b3*LOW_V +
a5*d4*b3*LOW_W*LOW_V;
   adbMWLV = a1*b3*d1 + a5*d1*b3*MED_W + a1*d4*b3*LOW_V +
a5*d4*b3*MED_W*LOW_V;
   adbHWLV = a1*b3*d1 + a5*d1*b3*HIGH_W + a1*d4*b3*LOW_V +
a5*d4*b3*HIGH_W*LOW_V;
   adbLWMV = a1*b3*d1 + a5*d1*b3*LOW_W + a1*d4*b3*MED_V +
a5*d4*b3*LOW_W*MED_V;
   adbMWMV = a1*b3*d1 + a5*d1*b3*MED_W + a1*d4*b3*MED_V +
a5*d4*b3*MED_W*MED_V;
   adbHWMV = a1*b3*d1 + a5*d1*b3*HIGH_W + a1*d4*b3*MED_V +
a5*d4*b3*HIGH_W*MED_V;
   adbLWHV = a1*b3*d1 + a5*d1*b3*LOW_W + a1*d4*b3*HIGH_V +
a5*d4*b3*LOW_W*HIGH_V;
   adbMWHV = a1*b3*d1 + a5*d1*b3*MED_W + a1*d4*b3*HIGH_V +
a5*d4*b3*MED_W*HIGH_V;
   adbHWHV = a1*b3*d1 + a5*d1*b3*HIGH_W + a1*d4*b3*HIGH_V +
a5*d4*b3*HIGH_W*HIGH_V;

! Now calc total indirect effects for each value of W and V
   TI_LWLV = a1b1LWLV + a2b2 + a3b3 + a2d2b3 + adbLWLV;
   TI_MWLV = a1b1MWLV + a2b2 + a3b3 + a2d2b3 + adbMWLV;
```

```
   TI_HWLV = a1b1HWLV + a2b2 + a3b3 + a2d2b3 + adbHWLV;
   TI_LWMV = a1b1LWMV + a2b2 + a3b3 + a2d2b3 + adbLWMV;
   TI_MWMV = a1b1MWMV + a2b2 + a3b3 + a2d2b3 + adbMWMV;
   TI_HWMV = a1b1HWMV + a2b2 + a3b3 + a2d2b3 + adbHWMV;
   TI_LWHV = a1b1LWHV + a2b2 + a3b3 + a2d2b3 + adbLWHV;
   TI_MWHV = a1b1MWHV + a2b2 + a3b3 + a2d2b3 + adbMWHV;
   TI_HWHV = a1b1HWHV + a2b2 + a3b3 + a2d2b3 + adbHWHV;

! Now calc total effects for each value of W and V
   TOT_LWLV = TI_LWLV + cdash;
   TOT_MWLV = TI_MWLV + cdash;
   TOT_HWLV = TI_HWLV + cdash;
   TOT_LWMV = TI_LWMV + cdash;
   TOT_MWMV = TI_MWMV + cdash;
   TOT_HWMV = TI_HWMV + cdash;
   TOT_LWHV = TI_LWHV + cdash;
   TOT_MWHV = TI_MWHV + cdash;
   TOT_HWHV = TI_HWHV + cdash;

! Use loop plot to plot total indirect effect of X on Y for
low, med, high values of W
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

   PLOT(PTI_LWLV PTI_MWLV PTI_HWLV PTI_LWMV PTI_MWMV PTI_HWMV
   PTI_LWHV PTI_MWHV PTI_HWHV);

   LOOP(XVAL,1,5,0.1);

   PTI_LWLV = TI_LWLV*XVAL;
   PTI_MWLV = TI_MWLV*XVAL;
   PTI_HWLV = TI_HWLV*XVAL;
   PTI_LWMV = TI_LWMV*XVAL;
   PTI_MWMV = TI_MWMV*XVAL;
   PTI_HWMV = TI_HWMV*XVAL;
   PTI_LWHV = TI_LWHV*XVAL;
   PTI_MWHV = TI_MWHV*XVAL;
   PTI_HWHV = TI_HWHV*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```

Model 80: 1 mediator, predictor has non-linear effect on mediator and outcome

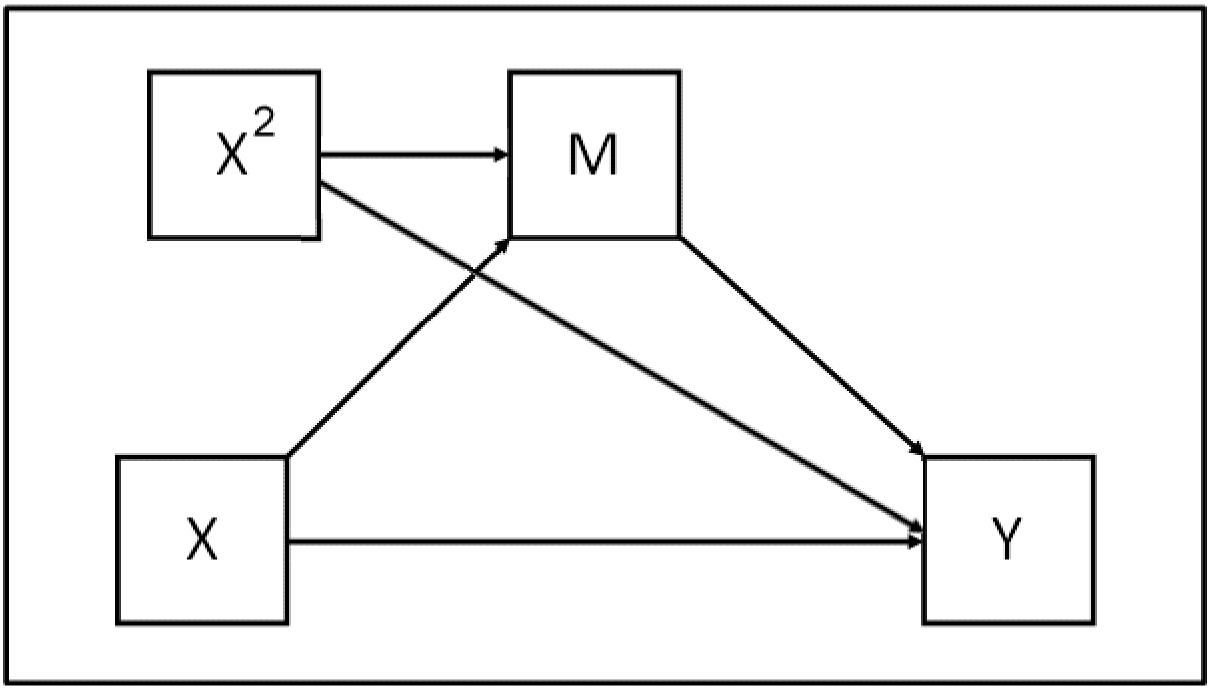
Example Variables: 1 predictor X, 1 mediator M, 1 outcome Y

Preliminary notes:

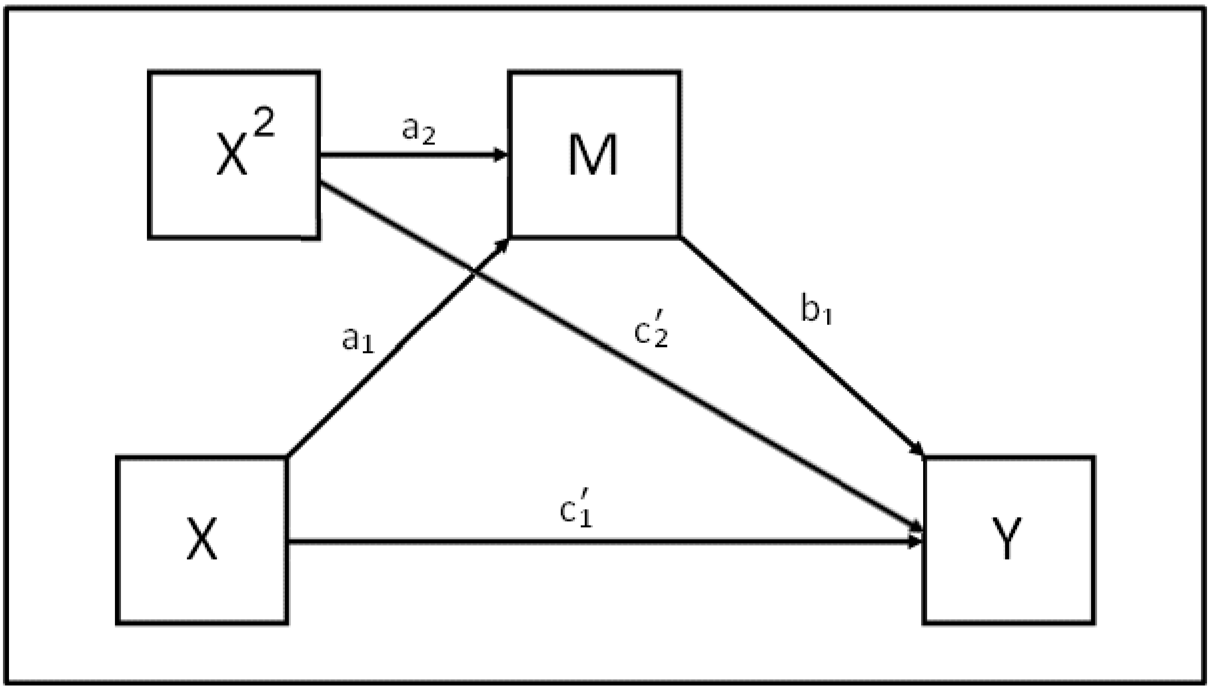
The code below assumes that

- The primary IV (variable X) is continuous or dichotomous
- Any moderators (variables W, V, Q, Z) are continuous, though the only adaptation required to handle dichotomous moderators is in the MODEL CONSTRAINT: and loop plot code - an example of how to do this is given in model 1b. Handling categorical moderators with > 2 categories is demonstrated in model 1d.
- Any mediators (variable M, or M1, M2, etc.) are continuous and satisfy the assumptions of standard multiple regression. An example of how to handle a dichotomous mediator is given in model 4c.
- The DV (variable Y) is continuous and satisfies the assumptions of standard multiple regression. An example of how to handle a dichotomous DV is given in model 1e (i.e. a moderated logistic regression) and in model 4d (i.e. an indirect effect in a logistic regression).

Model Diagram:



Statistical Diagram:



Model Equation(s):

Y = b0 + b1M + c1'X + c2'XX
M = a0 + a1X + a2XX

Algebra to calculate total, indirect and/or conditional effects by writing model as Y = a + bX:

Y = b0 + b1M + c1'X + c2'XX
M = a0 + a1X + a2XX

Hence... differentiating each equation to calculate the rates of change in the DV wrto the IV(s)

dY/dX = c1' + 2c2'X
dY/dM = b1
dM/dX = a1 + 2a2X

Hence... multiplying the relationships between X and M, and M and Y to get the indirect effect:

Instantaneous Indirect Effect (IIE) of X on Y:

(a1 + 2a2X)b1

And we also have the... Instantaneous Direct Effect (IDE) of X on Y:

c1' + 2c2'X

Mplus code for the model:

```
! Predictor variable(s) - X, XX
! Mediator variable(s) – M
! Moderator variable(s) - none
! Outcome variable - Y

USEVARIABLES = X XX M Y;

ANALYSIS:
   TYPE = GENERAL;
   ESTIMATOR = ML;
   BOOTSTRAP = 10000;

! In model statement name each path using parentheses

MODEL:
   Y ON M (b1);

   Y ON X (cdash1);    ! direct effect of X on Y
   Y ON XX (cdash2);    ! direct effect of X on Y

   M ON X (a1);
   M ON XX (a2);

! Use model constraint to calculate instantaneous indirect and
direct effects
! at different values of X

MODEL CONSTRAINT:
   NEW(LOW_X MED_X HIGH_X
   IIE_LOWX IIE_MEDX IIE_HIX
   IDE_LOWX IDE_MEDX IDE_HIX);

    LOW_X = #LOWX;    ! replace #LOWX in the code with your
chosen low value of X
    MED_X = #MEDX;    ! replace #MEDX in the code with your
chosen medium value of X
    HIGH_X = #HIGHX;    ! replace #HIGHX in the code with your
chosen high value of X

! Calc instantaneous indirect effects for low, medium, high
values of X

    IIE_LOWX = (a1 + 2*a2*LOW_X)*b1;
    IIE_MEDX = (a1 + 2*a2*MED_X)*b1;
    IIE_HIX = (a1 + 2*a2*HIGH_X)*b1;

! Calc instantaneous direct effects for low, medium, high
values of X
```

```
    IDE_LOWX = cdash1 + 2*cdash2*LOW_X;
    IDE_MEDX = cdash1 + 2*cdash2*MED_X;
    IDE_HIX = cdash1 + 2*cdash2*HIGH_X;

! Use loop plot to plot instantaneous indirect effect of X on
Y
! NOTE - values of 1,5 in LOOP() statement need to be replaced
by
! logical min and max limits of predictor X used in analysis

    PLOT(IIEX);

    LOOP(XVAL,1,5,0.1);

    IIEX = (a1*b1 + 2*a2*b1*XVAL)*XVAL;

PLOT:
   TYPE = plot2;

OUTPUT:
   STAND CINT(bcbootstrap);
```