MethodsCore
FirstLevel Script Documentation
09/05/2012
11/19/2013 - Updated

Introduction

The FirstLevel script lets you specify a first level design. The script is as flexible as the SPM GUI, but will save you a lot of time. The FirstLevel Script assumes you've already created a MasterData file that describes the trials of your experiment. Once that is done, there are four major sections to fill out:

1. Image and subject information
2. MasterData file and condition information
3. User-specified regressor information
4. Contrast information

Additionally, there is an Advanced Section to the script, which only some options will be covered. There is a brief description of the advanced features in the script itself. If you want to use any of these features to the fullest extent, we recommend you contact MethodsCoreHelp@umich.edu for information. Filling in these four sections in the script is mostly self-explanatory, and there is extensive help in the script itself. Below we'll walk you through the steps that require more explanation.

Visual Tutorials

There are a series of video tutorials available online. It may be useful to view these while following along in this document. As of 11/19/13, parts of these videos are no longer relevant.

1 Introduction	http://youtu.be/FHWJgJFLMU8?hd=1
2 MasterData File	http://youtu.be/6OoV_SPkzSU?hd=1
3 Subjects & Runs	http://youtu.be/epfNW4u-z9c?hd=1
4 Conditions, Onsets, & Durations	http://youtu.be/xsLuvXqwZy4?hd=1
5 Parametric Regressors	http://youtu.be/SbXZ0CpxerE?hd=1
6 User-Specified Regressors	http://youtu.be/D5kw4RJY1GU?hd=1
7 Contrasts	http://youtu.be/EPSB2FObnyY?hd=1
8 Conclusion	http://youtu.be/CrRaQELaKNw?hd=1

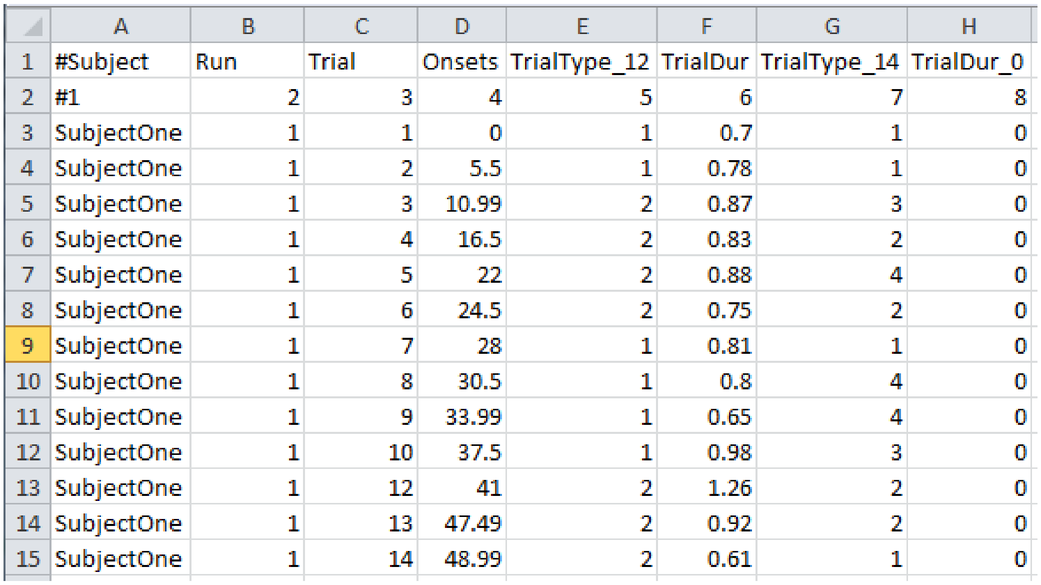
Getting Started

Copy the FirstLevel_template script from your Methods Core directory into your local experiment directory. This is the file you'll be editing.

Part 1: MasterData Files

The master data file is a CSV file that contains all the information about your experiment. All the information normally specified in the SPM GUI subject-by-subject and condition-by-condition is now specified in a single file. This makes analysis easier, more flexible, and more transparent. Its syntax has been revamped, so previous master data files are incompatible without minor modifications. The master data file can still be a trial-by-trial exhaustive list of every trial in an experiment and specifying the master data file in this manner will be discussed first. Additionally, the master data file can hold pointers to subject specific data files which can hold pointers to run specific files. Creating a master data file with pointers will be discussed second.

Trial-by-trial data file:

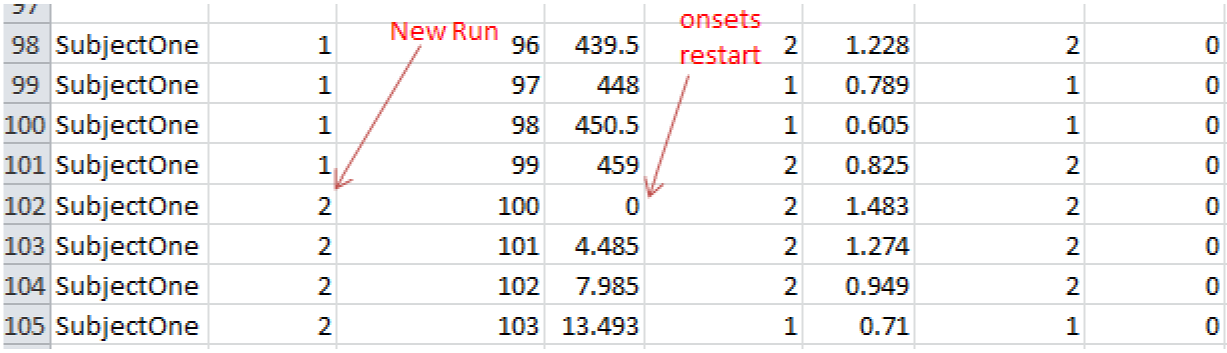


	A	B	C	D	E	F	G	H
1	#Subject	Run	Trial	Onsets	TrialType_12	TrialDur	TrialType_14	TrialDur_0
2	#1	2	3	4	5	6	7	8
3	SubjectOne	1	1	0	1	0.7	1	0
4	SubjectOne	1	2	5.5	1	0.78	1	0
5	SubjectOne	1	3	10.99	2	0.87	3	0
6	SubjectOne	1	4	16.5	2	0.83	2	0
7	SubjectOne	1	5	22	2	0.88	4	0
8	SubjectOne	1	6	24.5	2	0.75	2	0
9	SubjectOne	1	7	28	1	0.81	1	0
10	SubjectOne	1	8	30.5	1	0.8	4	0
11	SubjectOne	1	9	33.99	1	0.65	4	0
12	SubjectOne	1	10	37.5	1	0.98	3	0
13	SubjectOne	1	12	41	2	1.26	2	0
14	SubjectOne	1	13	47.49	2	0.92	2	0
15	SubjectOne	1	14	48.99	2	0.61	1	0

Figure 1

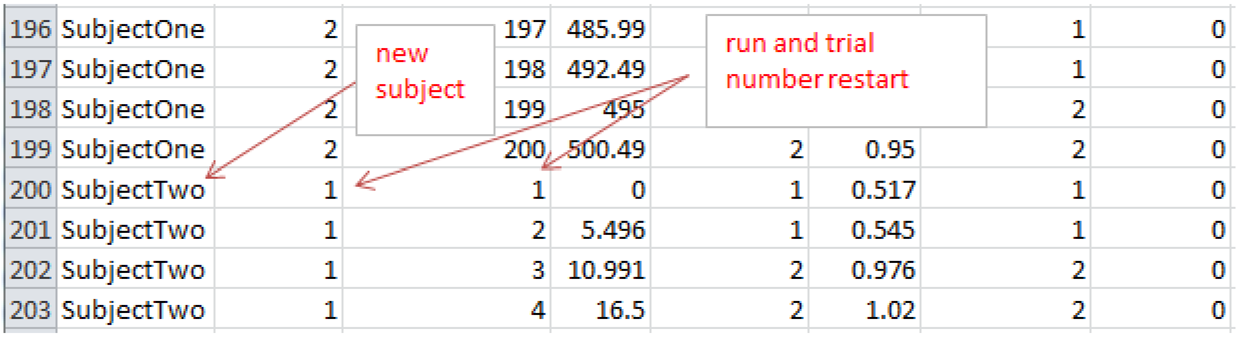
Look at Figure 1. I have two header rows here: row 1 identifies the type of column and row 2 provides the column number (you'll see why this is useful a bit later). Notice the first column entries in the headers begin with the pound sign '#'. When placed at the beginning of a line, the '#' character marks rows to ignore when reading data files (previously this was done by the MasterDataSkipRows variable in the first level template file which has been removed). The remaining content of the master data file is a trial-by-trial exhaustive list of every trial in the experiment. In the above spreadsheet, I have listed for each trial: the Subject (col 1), Run (col 2), Trial Number (col 3). I also have information for each trial about its Onset, Condition, and Duration but I'll get to that a bit later.

The MasterData file can be one long list. When I get to the end of a run, I start the next run and continue listing trials. The onsets restart because SPM wants to know the onsets relative to the start of a run.



98	SubjectOne	1	96	439.5	2	1.228	2	0
99	SubjectOne	1	97	448	1	0.789	1	0
100	SubjectOne	1	98	450.5	1	0.605	1	0
101	SubjectOne	1	99	459	2	0.825	2	0
102	SubjectOne	2	100	0	2	1.483	2	0
103	SubjectOne	2	101	4.485	2	1.274	2	0
104	SubjectOne	2	102	7.985	2	0.949	2	0
105	SubjectOne	2	103	13.493	1	0.71	1	0

If you start a new subject, then you list the new subject number, and restart run number and trial number (and onsets).



196	SubjectOne	2		197	485.99				1	0
197	SubjectOne	2		198	492.49				1	0
198	SubjectOne	2		199	495				2	0
199	SubjectOne	2		200	500.49	2	0.95		2	0
200	SubjectTwo	1		1	0	1	0.517		1	0
201	SubjectTwo	1		2	5.496	1	0.545		1	0
202	SubjectTwo	1		3	10.991	2	0.976		2	0
203	SubjectTwo	1		4	16.5	2	1.02		2	0

Strings can now be present in the master data file except in the run, trial, onset, and duration columns. If they are present in these columns, an error will occur when reading the master data file. Strings are also used to identify subjects now. Of course you can have a string of numbers to identify subjects.

Runs, Conditions, onsets, and durations

For each trial we first assign it to a run. This trial should occur in that run during the experiment. In the master data file, the run numbers should be chosen such that they correspond with the runs to include vector in the SubjDir variable declared in the template file. The SubjDir variable will be discussed in more detail later.

For each trial, we want to be able to assign to it a ***Condition***, an ***Onset***, and a ***Duration***. Recall that for an SPM analysis this information is required for every single trial. Assigning the ***Onset*** and ***Duration*** are easy – they are simply listed in Figure 1 in columns 4 and 6.

The ***Condition*** assignment is done with a numerical method. First, I assign each condition a number starting with 1. So let's say I have two conditions, Congruent and Incongruent. Then I would assign Congruent as 1 and Incongruent as 2. Next, for each trial, I assign it the appropriate condition using these numbers. For example, in Figure 1, column 5, I assigned every trial a 1 or a 2 based on whether it is Congruent or Incongruent. A bit later, we will 'point to' the relevant columns in the MasterData file as we set up our FirstLevel script.

Pointer data file:

With the new data file syntax, pointers to other data files can be included in another data file, which permit the usage of multiple data files for a first level analysis. Users can now store run information at the subject and run level (Figure 2). The '@' symbol indicates a line in a data file points to another data file (figure 3). Immediately after the '@' symbol comes the full file path to the other data file. All data files need to have the subject, run, condition, onset, and duration information in the same columns; otherwise, the wrong information will probably be used to generate the first level results. Data files can store both data and pointers if the user desires to do such a thing.

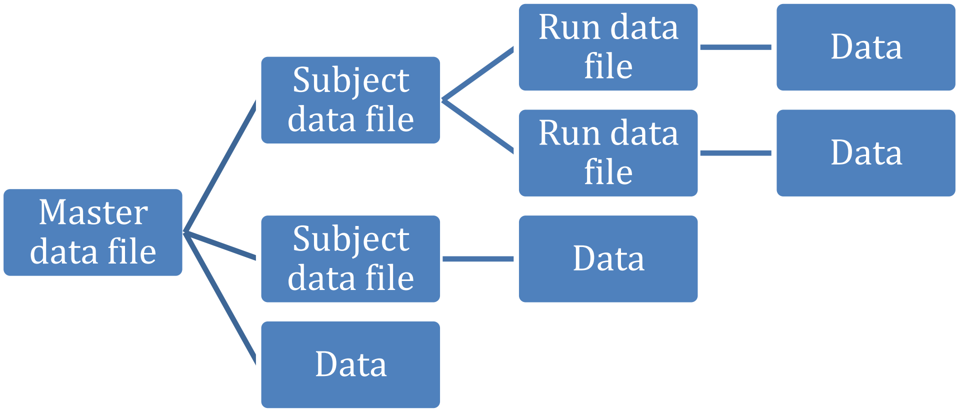
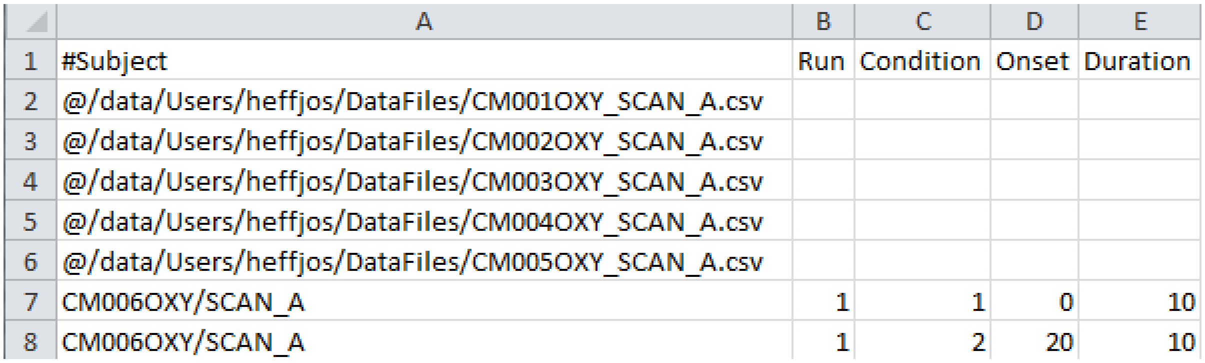


Figure 2. Schematic outlining pointer data file method.



	A	B	C	D	E
1	#Subject	Run	Condition	Onset	Duration
2	@/data/Users/heffjos/DataFiles/CM001OXY_SCAN_A.csv				
3	@/data/Users/heffjos/DataFiles/CM002OXY_SCAN_A.csv				
4	@/data/Users/heffjos/DataFiles/CM003OXY_SCAN_A.csv				
5	@/data/Users/heffjos/DataFiles/CM004OXY_SCAN_A.csv				
6	@/data/Users/heffjos/DataFiles/CM005OXY_SCAN_A.csv				
7	CM006OXY/SCAN_A	1	1	0	10
8	CM006OXY/SCAN_A	1	2	20	10

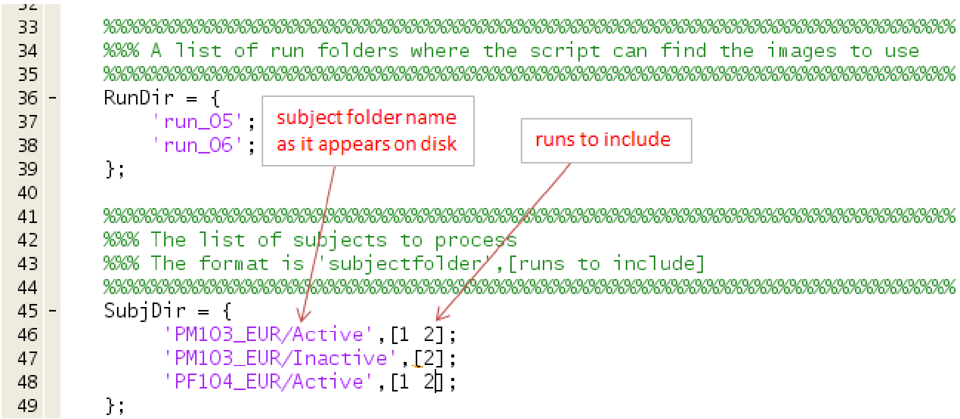
Figure 3. An example data file using pointer syntax (rows 2-6). Normal data is also included (rows 7-8), so this is a mixed data file.

Part 2: Setting up the FirstLevel_template Script

You specify a lot of experiment-related information in the FirstLevel script. Most of this should be self-explanatory and there is help in the script itself.

One important issue is setting up paths. You set up the paths to a lot of things (path to functional images, output path, MasterData file, regressor file if you have one, etc…). Setting up paths is done through a standardized 'Path Template' system. See the **Path Template Documentation** in the root directory of the Methods Core repository for help on this. We'll be using this method for setting up paths in pretty much all Methods Core software.

Setting up Subjects and Runs



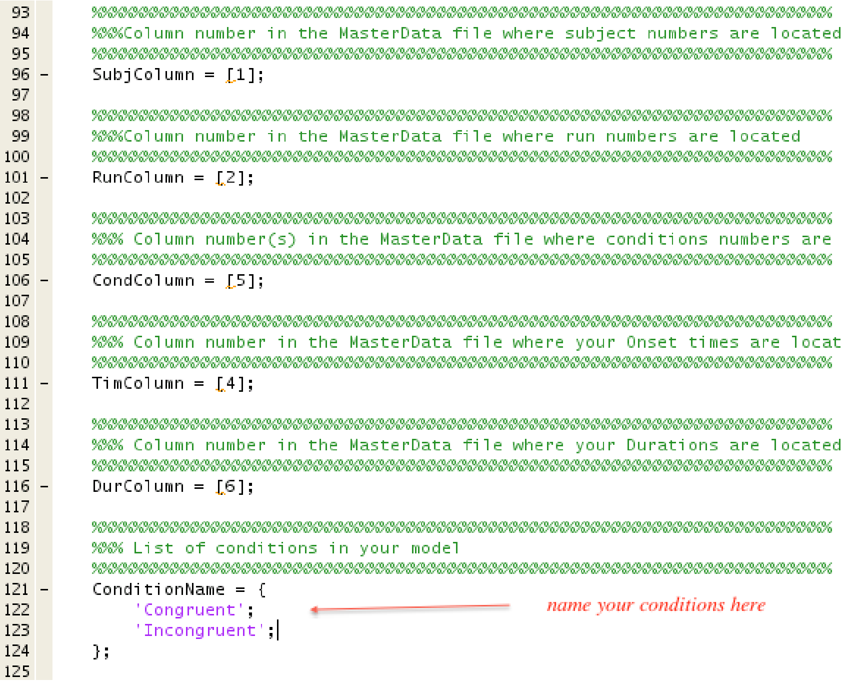
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% A list of run folders where the script can find the images to use
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RunDir = {
    'run_05';
    'run_06';
};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% The list of subjects to process
%%% The format is 'subjectfolder',[runs to include]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SubjDir = {
     'PM103_EUR/Active',[1 2];
     'PM103_EUR/Inactive',[2];
     'PF104_EUR/Active',[1 2];
};
```

RunDir lists the run names as they appear on the disk. SubjDir lists the subjects to include in the analysis. It has two columns that are explained above. Notice that 'runs to include' (i.e., column 2 of SubjDir) indexes the rows of RunDir. So in the above example, we are omitting run_05 for subject 'PM103_EUR/Inactive'.

Setting up conditions, onsets, durations, etc..

In the FirstLevel_template Script, you 'point to' the relevant columns the MasterData file we set up earlier (see Figure 2 below). The script will then 'find' the information from the MasterData file. It will go to just the subjects, runs you tell it to, and will ignore the rest. Notice you also name your conditions, and you need to make sure the order of names corresponds to the order of numbers you used in the MasterData File.

But let's say I change my mind and want to model my data with four conditions: Congruent, Incongruent, Non-response, and Errors. Then I create a new condition column in the MasterData file with the numbers 1 through 4 that represent these conditions. This is column 7 in Figure 1. Then in the FirstLevel script (see Figure 4 below), I replace The CondColumn with 7, and list the new condition names in the ConditionName variable.



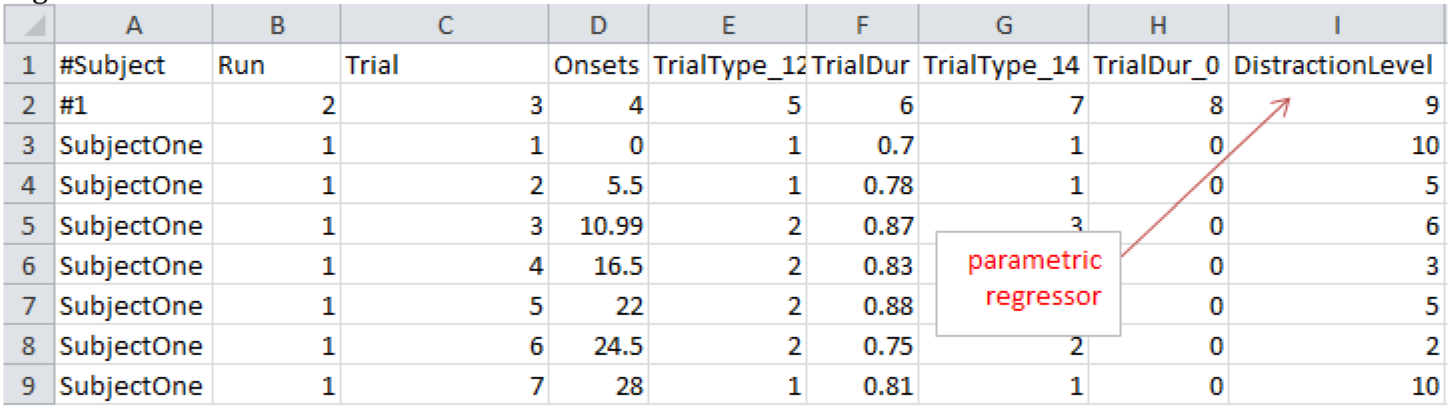
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Column number in the MasterData file where subject numbers are located
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SubjColumn = [1];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Column number in the MasterData file where run numbers are located
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RunColumn = [2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Column number(s) in the MasterData file where conditions numbers are
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CondColumn = [5];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Column number in the MasterData file where your Onset times are locat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TimColumn = [4];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Column number in the MasterData file where your Durations are located
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DurColumn = [6];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% List of conditions in your model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ConditionName = {
    'Congruent';
    'Incongruent';
};
```

name your conditions here

Figure 4

Parametric Regressors

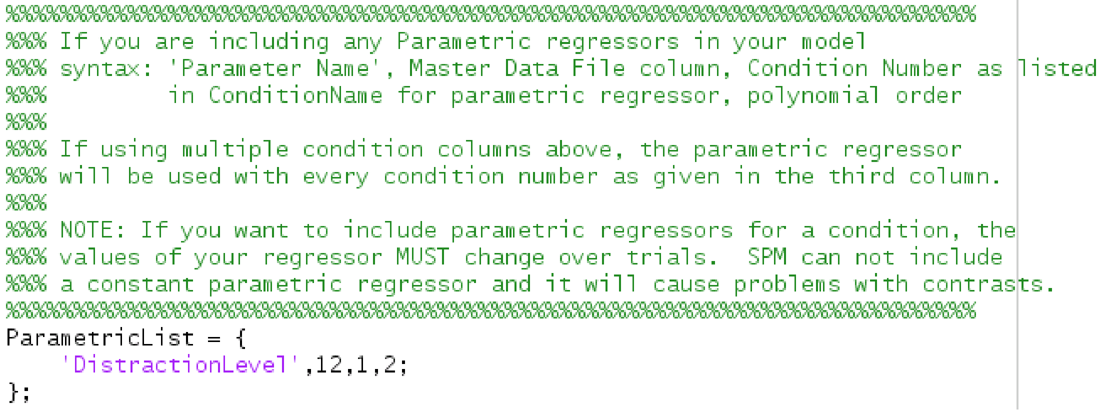
A parametric regressor is a regressor that modulates activity on a trial-by-trial basis. Adding a parametric regressor is easy. Just add a new column in the MasterData file that specifies the value the parametric regressor takes for each trial.



	A	B	C	D	E	F	G	H	I
1	#Subject	Run	Trial	Onsets	TrialType_12	TrialDur	TrialType_14	TrialDur_0	DistractionLevel
2	#1	2	3	4	5	6	7	8	9
3	SubjectOne	1	1	0	1	0.7	1	0	10
4	SubjectOne	1	2	5.5	1	0.78	1	0	5
5	SubjectOne	1	3	10.99	2	0.87	3	0	6
6	SubjectOne	1	4	16.5	2	0.83		0	3
7	SubjectOne	1	5	22	2	0.88		0	5
8	SubjectOne	1	6	24.5	2	0.75	2	0	2
9	SubjectOne	1	7	28	1	0.81	1	0	10

parametric regressor

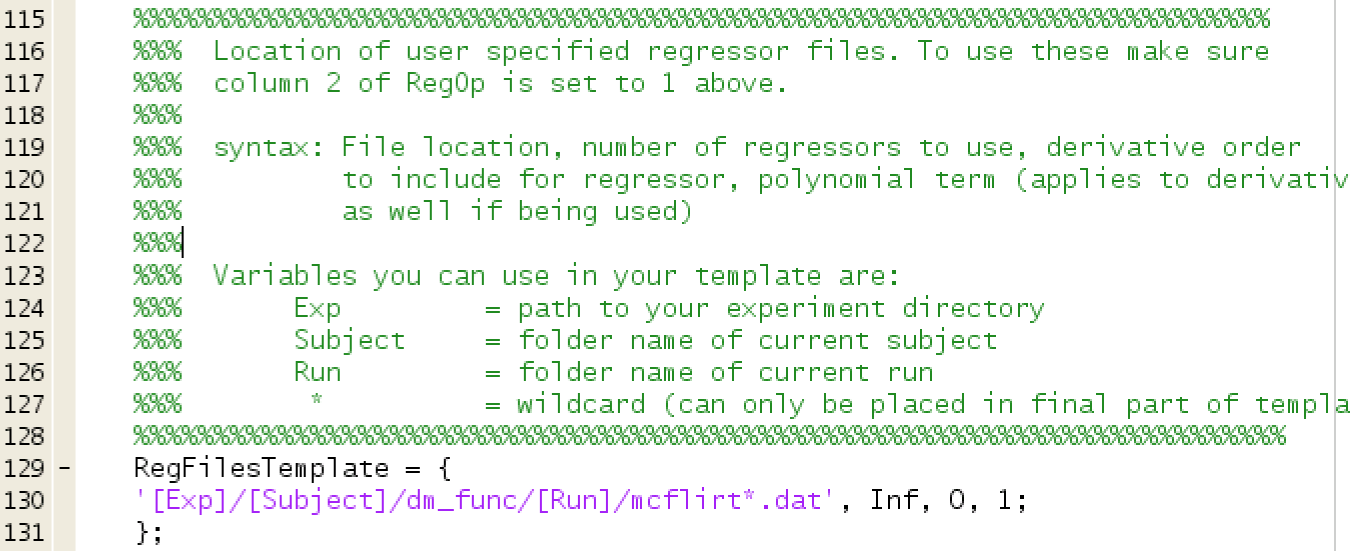
Then you list the name of the parametric regressor, its associated column in the FirstLevel script, the corresponding condition number, and the polynomial order (typically this will be 1).



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% If you are including any Parametric regressors in your model
%%% syntax: 'Parameter Name', Master Data File column, Condition Number as listed
%%%         in ConditionName for parametric regressor, polynomial order
%%%
%%% If using multiple condition columns above, the parametric regressor
%%% will be used with every condition number as given in the third column.
%%%
%%% NOTE: If you want to include parametric regressors for a condition, the
%%% values of your regressor MUST change over trials.  SPM can not include
%%% a constant parametric regressor and it will cause problems with contrasts.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ParametricList = {
    'DistractionLevel',12,1,2;
};
```

User-Specified Regressors

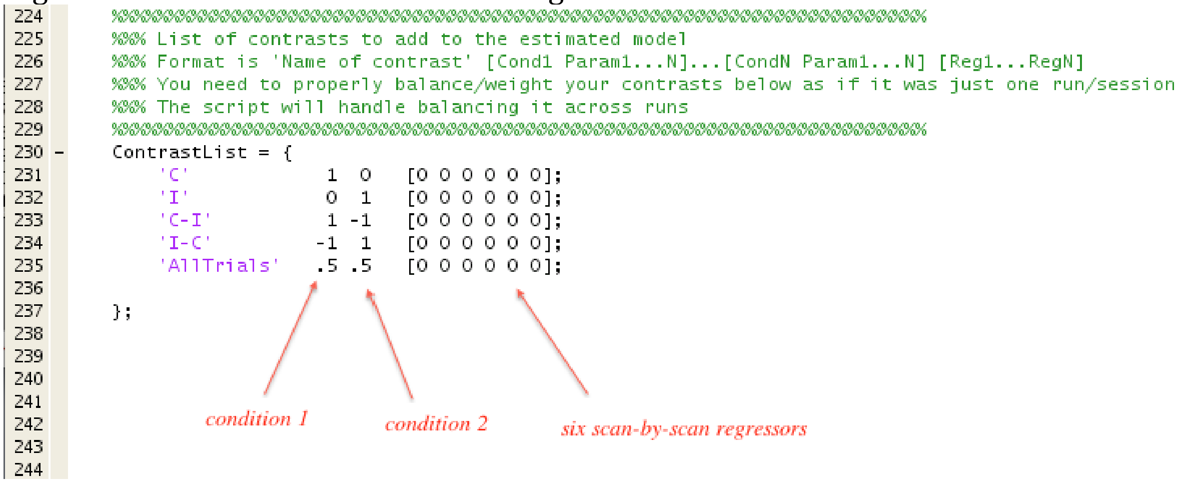
A user-specified regressor (the terminology is from SPM) is a regressor that modulates activity on a scan-by-scan basis. The most common scan-by-scan regressors are your motion regressors. So the FirstLevel script lets you specify scan-by-scan regressors by specifying a variable called RegFileTemplates. Each line of RegFileTemplates refers to a single regressor file, followed by a number indicating the number of columns you'd like to use from that file (use Inf to automatically use all the columns), followed by another number which indicates the derivative order to include for the regressor file (0 = no derivatives, 1 = first derivative, and so on), and final another number used to include the polynomial order for the file (1 = none, 2 = quadratic, 3 = cubic, and so on).   The polynomial order is also applied to any derivatives calculated.



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Location of user specified regressor files. To use these make sure
%%%  column 2 of RegOp is set to 1 above.
%%%
%%%  syntax: File location, number of regressors to use, derivative order
%%%          to include for regressor, polynomial term (applies to derivatives
%%%          as well if being used)
%%%
%%%  Variables you can use in your template are:
%%%       Exp          = path to your experiment directory
%%%       Subject      = folder name of current subject
%%%       Run          = folder name of current run
%%%         *          = wildcard (can only be placed in final part of template)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RegFilesTemplate = {
'[Exp]/[Subject]/dm_func/[Run]/mcflirt*.dat', Inf, 0, 1;
};
```

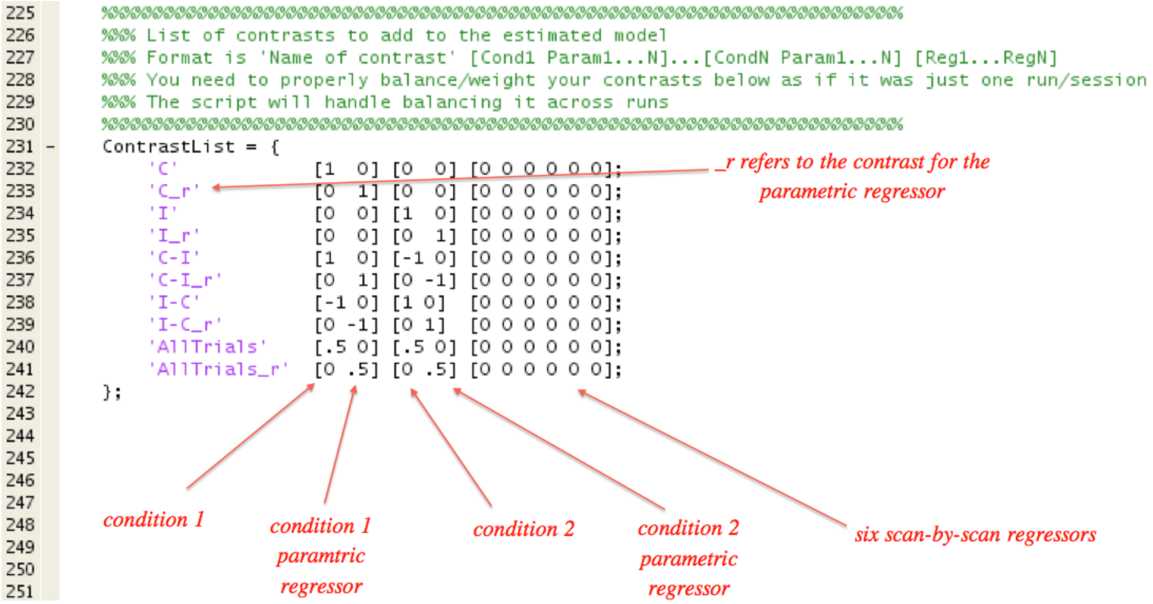
Contrasts

Let's say you had an experiment with two conditions, Congruent and Incongruent, and 6 user-specified regressors. Here is what the contrast might look like:



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% List of contrasts to add to the estimated model
%%% Format is 'Name of contrast' [Cond1 Param1...N]...[CondN Param1...N] [Reg1...RegN]
%%% You need to properly balance/weight your contrasts below as if it was just one run/session
%%% The script will handle balancing it across runs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ContrastList = {
    'C'            1  0   [0 0 0 0 0 0];
    'I'            0  1   [0 0 0 0 0 0];
    'C-I'          1 -1   [0 0 0 0 0 0];
    'I-C'         -1  1   [0 0 0 0 0 0];
    'AllTrials'   .5 .5   [0 0 0 0 0 0];

};
```

Let's say you now have 2 conditions and a single parametric regressor, and 6 user-specified regressors. Here is what the contrast might look like.



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% List of contrasts to add to the estimated model
%%% Format is 'Name of contrast' [Cond1 Param1...N]...[CondN Param1...N] [Reg1...RegN]
%%% You need to properly balance/weight your contrasts below as if it was just one run/session
%%% The script will handle balancing it across runs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ContrastList = {
    'C'            [1  0] [0  0] [0 0 0 0 0 0];
    'C_r'          [0  1] [0  0] [0 0 0 0 0 0];
    'I'            [0  0] [1  0] [0 0 0 0 0 0];
    'I_r'          [0  0] [0  1] [0 0 0 0 0 0];
    'C-I'          [1  0] [-1 0] [0 0 0 0 0 0];
    'C-I_r'        [0  1] [0 -1] [0 0 0 0 0 0];
    'I-C'          [-1 0] [1 0]  [0 0 0 0 0 0];
    'I-C_r'        [0 -1] [0 1]  [0 0 0 0 0 0];
    'AllTrials'    [.5 0] [.5 0] [0 0 0 0 0 0];
    'AllTrials_r'  [0 .5] [0 .5] [0 0 0 0 0 0];
};
```

_r refers to the contrast for the parametric regressor
condition 1
condition 1 paramtric regressor
condition 2
condition 2 parametric regressor
six scan-by-scan regressors

Brackets (i.e., [and]) that are placed around the condition/parametric regressor pair and also around the six scan-by-scan regressors are not for cosmetic purposes. The script uses the brackets to parse your ContrastList so do not omit the brackets!

Some other rules to keep in mind are (these are for most standard analyses, and there are exceptions):

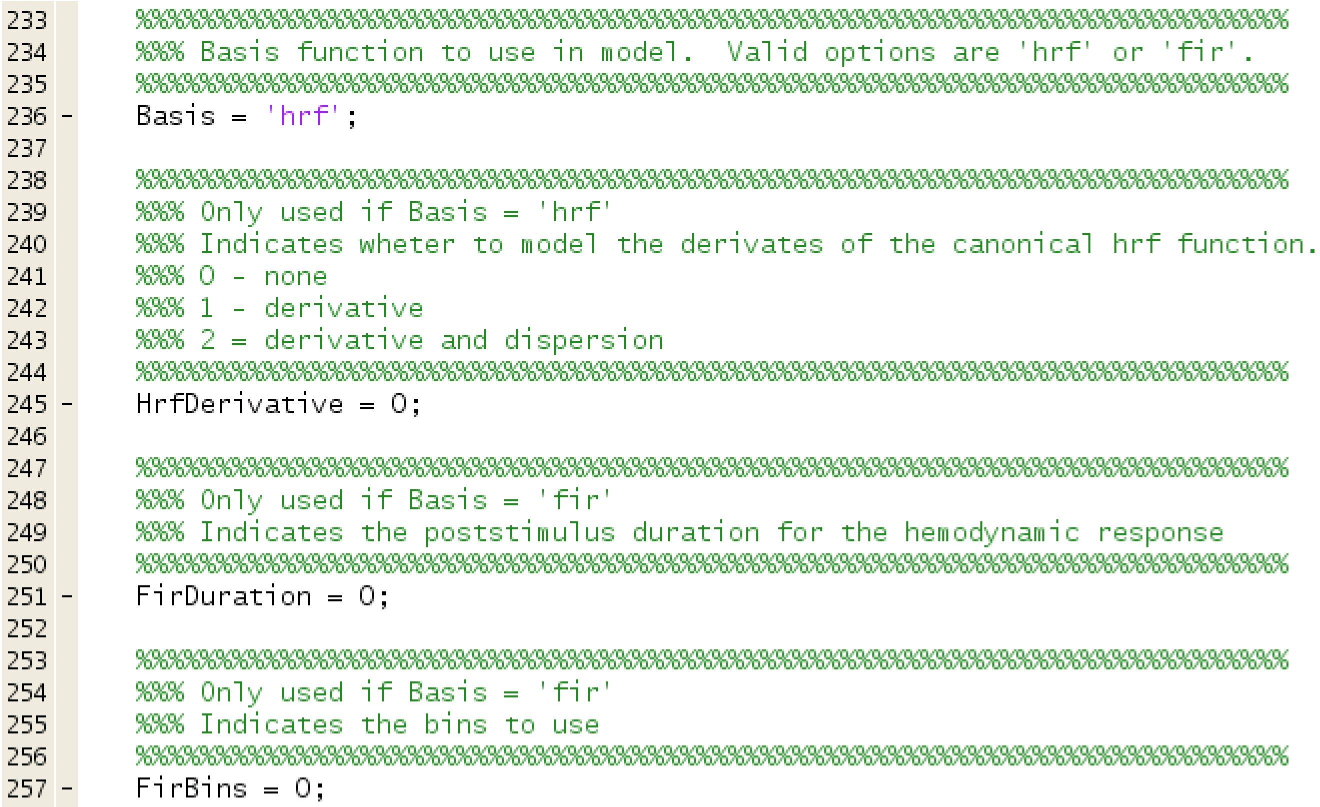
- Contrasts against an implicit baseline (e.g., 'C') – positive numbers sum to 1, no negative numbers
- Contrasts of two conditions against each other (e.g., 'I-C') - positive numbers sum to 1, negative numbers sum to 1

If you follow these rules, your estimated betas will correspond to percent signal change, which is a good thing. The script helps you follow these rules by balancing your numbers across runs. Here is how it works. You specify the contrasts as if there were only one run and you do not include any trailing zeros for the SPM-created run-related regressor. The script will automatically expand/rescale the numbers for the number of runs in your experiment. For example, if there are three runs, all the numbers in the contrasts above would be multiplied by 1/3, and additional trailing zeros for the SPM-created run-related regressors (one per run) would be automatically be created.

To repeat, there is also an Advanced Section, with a LOT of other features. For example, you can have run specific contrasts. You reset SPM defaults. You can drop conditions that appear just a few times. And many others. If you want to use any of these features, we recommend you contact MethodsCoreHelp@umich.edu for information.

Advanced: Basis functions

In this context, basis functions are the assumed brain response caused by an external stimulus (i.e. a trial in an experiment). The advanced section gives the user flexibility choosing which basis functions to use in a first level analysis (Figure 5). With the 'Basis' variable, you can choose between using the canonical HRF or finite impulse response (FIR) basis functions. When using the canonical HRF, you can additionally model the time and dispersion derivatives using the 'HrfDerivative' variable. If you do model the derivatives, this will affect how you setup your contrasts. Example contrasts will be given later.



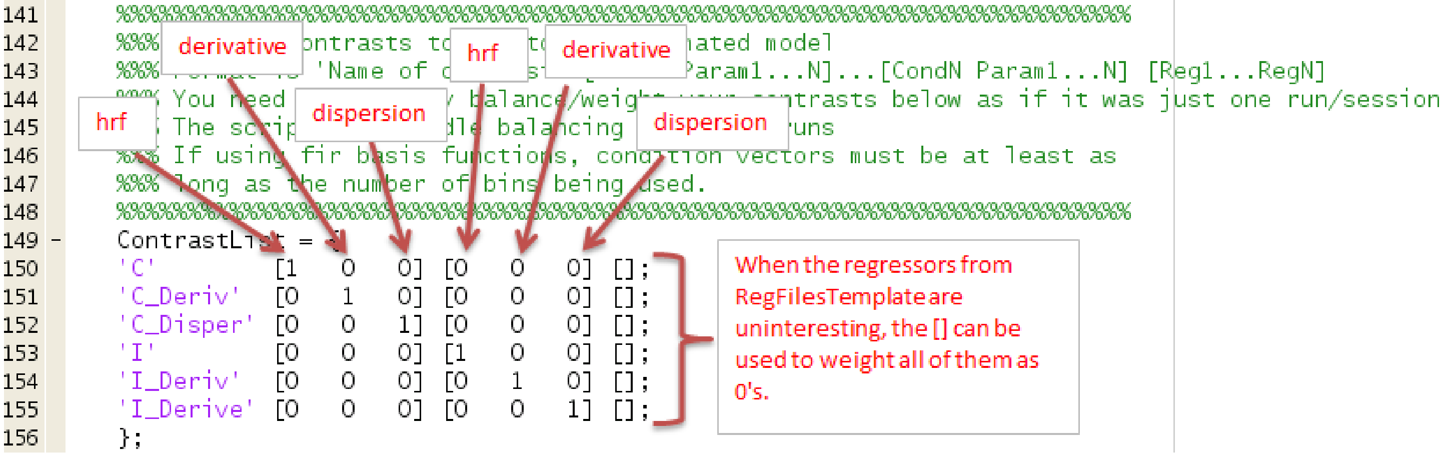
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Basis function to use in model.  Valid options are 'hrf' or 'fir'.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Basis = 'hrf';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Only used if Basis = 'hrf'
%%% Indicates wheter to model the derivates of the canonical hrf function.
%%% 0 - none
%%% 1 - derivative
%%% 2 = derivative and dispersion
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
HrfDerivative = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Only used if Basis = 'fir'
%%% Indicates the poststimulus duration for the hemodynamic response
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
FirDuration = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Only used if Basis = 'fir'
%%% Indicates the bins to use
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
FirBins = 0;
```

Figure 5. The variables used for setting basis functions in the advanced section.

To use an FIR model in the first level, set 'Basis' to 'fir'. The FIR model is commonly used to estimate a condition time course in an experiment. The post-stimulus duration is controlled by 'FirDuration' and the number of basis functions is controlled by the 'FirBins' variable. If we wanted to estimate a 30 second post-stimulus time course with 15 FIR basis functions, we set FirDuation=30 and FirBins=15.

Advanced: Contrasts

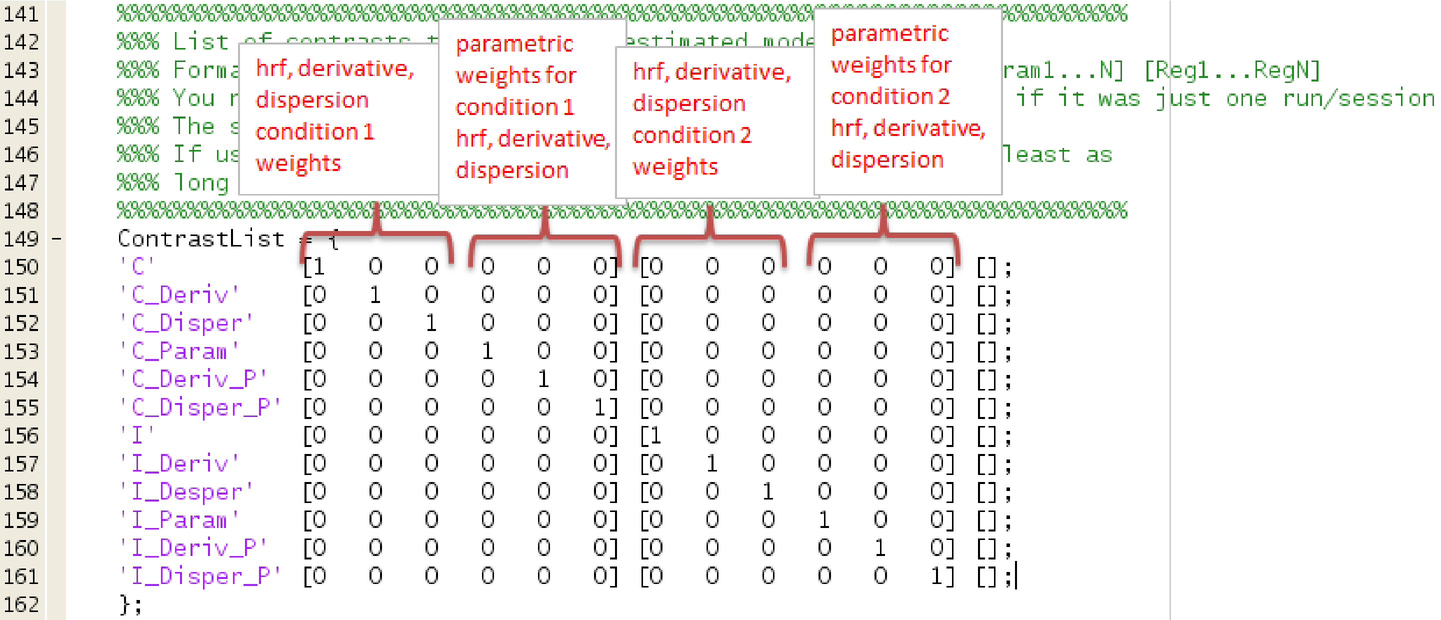
Let's say we have 2 conditions and are using the 'hrf' basis with the derivative and dispersion regressors.



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%            ontrasts t      t            ated model
%%%                 'Name of c     s           Param1...N]...[CondN Param1...N] [Reg1...RegN]
     You need                  balance/weig          ntrasts below as if it was just one run/session
   The scrip               dle balancing            uns
%%% If using fir basis functions, condition vectors must be at least as
%%% long as the number of bins being used.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ContrastList = {
'C'         [1   0   0]  [0   0   0]  [];
'C_Deriv'   [0   1   0]  [0   0   0]  [];
'C_Disper'  [0   0   1]  [0   0   0]  [];
'I'         [0   0   0]  [1   0   0]  [];
'I_Deriv'   [0   0   0]  [0   1   0]  [];
'I_Derive'  [0   0   0]  [0   0   1]  [];
};
```

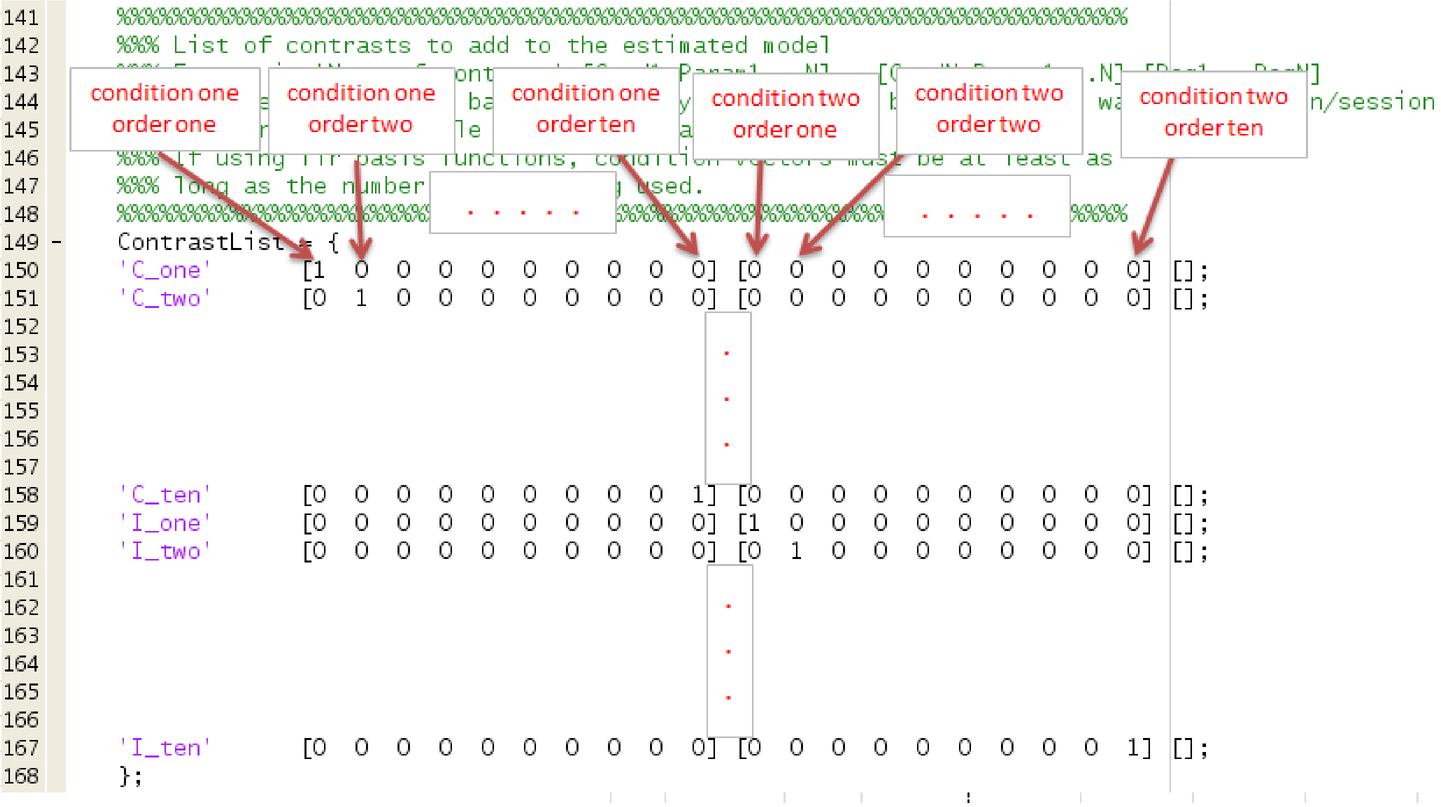
When the regressors from RegFilesTemplate are uninteresting, the [] can be used to weight all of them as 0's.

Let's say we have 2 conditions, are using the 'hrf' basis with the derivative and dispersion regressors, and are using parametric regressors for both conditions. The contrasts will be similar to above, except three additional columns are added for both condition vectors for the parametric regressors.


hrf, derivative, dispersion condition 1 weights
parametric weights for condition 1 hrf, derivative, dispersion
hrf, derivative, dispersion condition 2 weights
parametric weights for condition 2 hrf, derivative, dispersion

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% List of contrasts t                      estimated mode
%%% Forma                                                        ram1...N] [Reg1...RegN]
%%% You r                                                          if it was just one run/session
%%% The s
%%% If us                                                        least as
%%% long
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ContrastList = {
'C'           [1  0  0  0  0  0] [0  0  0  0  0  0] [];
'C_Deriv'     [0  1  0  0  0  0] [0  0  0  0  0  0] [];
'C_Disper'    [0  0  1  0  0  0] [0  0  0  0  0  0] [];
'C_Param'     [0  0  0  1  0  0] [0  0  0  0  0  0] [];
'C_Deriv_P'   [0  0  0  0  1  0] [0  0  0  0  0  0] [];
'C_Disper_P'  [0  0  0  0  0  1] [0  0  0  0  0  0] [];
'I'           [0  0  0  0  0  0] [1  0  0  0  0  0] [];
'I_Deriv'     [0  0  0  0  0  0] [0  1  0  0  0  0] [];
'I_Desper'    [0  0  0  0  0  0] [0  0  1  0  0  0] [];
'I_Param'     [0  0  0  0  0  0] [0  0  0  1  0  0] [];
'I_Deriv_P'   [0  0  0  0  0  0] [0  0  0  0  1  0] [];
'I_Disper_P'  [0  0  0  0  0  0] [0  0  0  0  0  1] [];
};
```

Now let's say we have 2 conditions, are using the 'fir' basis function with an order equal to 10. We will have 10 weights for each condition.



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% List of contrasts to add to the estimated model
                                                ont          Param1...N]  [C       .N]  [Reg1...RegN]
                  e                             ba                     y     b        wa      n/session
                  r                             le                     a
%%% If using fir basis functions, condition vectors must be at least as
%%% long as the number              used.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ContrastList = {
'C_one'       [1  0  0  0  0  0  0  0  0  0] [0  0  0  0  0  0  0  0  0  0] [];
'C_two'       [0  1  0  0  0  0  0  0  0  0] [0  0  0  0  0  0  0  0  0  0] [];






'C_ten'       [0  0  0  0  0  0  0  0  0  1] [0  0  0  0  0  0  0  0  0  0] [];
'I_one'       [0  0  0  0  0  0  0  0  0  0] [1  0  0  0  0  0  0  0  0  0] [];
'I_two'       [0  0  0  0  0  0  0  0  0  0] [0  1  0  0  0  0  0  0  0  0] [];






'I_ten'       [0  0  0  0  0  0  0  0  0  0] [0  0  0  0  0  0  0  0  0  1] [];
};
```