

How to do repeated measure ANOVA in MATLAB? (Written by Parisa Abedi, updated October 2018)

A simple way is to use `anovan` (n-way ANOVA) in MATLAB. To learn about how it works you can look at following link: <http://www.mathworks.com/help/stats/anovan.html>.

Assume that we have 2 participants for the experiment (S1 and S2) reaching to 2 targets positioned in a 10cm radius circle with angle 45 and 135deg from horizontal lane. Participants will reach to these targets in three head roll conditions: 30CCW (represented as -30), 0, 30CW (represented as +30). We calculate their reach error in degree for our analysis. Then we have 3 tables:

- 1- Errors: [1.02 2.5 3.65]
- 2- Target positions: [45 135]
- 3- HR conditions : [-30 0 30]

I should note that I assumed that each subjects reached to each target one per different HR conditions. If you have repetition in your experiment you can find the error value for each condition by averaging among all the values.

Since we have more than one measurement for each participant, we should perform repeated measure ANOVA. To do so in MATLAB, we should add the subject number as another factor to our n-way anova and set it as random factor.

The following figure is an example of organizing your data:

Directional Error	Target	HR condition	Subject number
3.025	60	-30	1
2.054	120	-30	1
0.25	60	0	1
0.55	120	0	1
4.01	60	30	1
1.98	120	30	1
4.001	60	-30	2
3.05	120	-30	2
0.56	60	0	2
1.01	120	0	2
2.058	60	30	2
3.065	120	30	2

Then the MATLAB command could be:

```
[p,table,stats] = anovan(Directional_Error,{Target,HRcondition,Subjects},...  
                        'random',3,'varnames', {'Targets','HR','Subjects'});
```

This way you have the general ANOVA working; however, if you use the 'stats' output for posthoc analysis you get into trouble. In MATLAB, if you use 'anovan' results for repeated measure ANOVA,

'multcompare' ignores the repeated measure set up. Therefore, in the reminder of the code, I will share how to run it with 'ranova' and then how to properly run it for 3-way repeated measure ANOVA.

In order to have a good understanding of the 'ranova' please read the official MATLAB description. This tutorial is intended for more specific design; i.e. only having within independent variables. In this example I will explain 3-way ANOVA as you can easily adapt simpler, 2-way ANOVA, or more complicated ones using this example.

Like previous one, I will explain the whole procedure using an example:

The experiment is to measure the biases in obstacle avoidance trajectories created due to tilting the head. We measured, 18 subjects in three different head rolls: 0,30CW, and 30CCW. In addition, we have two different visual feedback conditions (with and without VF) and 5 different obstacle positions (O1-O5).

The first step is easily to organize the data properly in a table format:

```
Biases = rand(18,3*5*2); % subjects, HRs*obstacle pos.*VFs

varNames = cell(3*5*2,1);
for i = 1 : 3*5*2
    v = strcat('V',num2str(i));
    varNames{i,1} = v;
end
% Create a table storing the responses
tbiases = array2table(Biases, 'VariableNames',varNames);
```

Then, create a table reflecting the within subject factors:

```
% Create a table reflecting the within subject factors
HRs = cell(3*5*2,1); % head roll conditions
VFs = cell(3*5*2,1); % Visual feedback conditions
OPs = cell(3*5*2,1); % Obstacle Positions

% Assigning the values to the parameters based on the data sorting
c1 = cell(1,1); c1{1} = 'Y'; c1 = repmat(c1,15,1); VFs(1: 15,1) = c1;
c1 = cell(1,1); c1{1} = 'N'; c1 = repmat(c1,15,1); VFs(16: end,1) = c1;

c1 = cell(1,1); c1{1} = 'HR0'; c1 = repmat(c1,10,1); HRs(1:3:end,1) = c1;
c1 = cell(1,1); c1{1} = 'HRL'; c1 = repmat(c1,10,1); HRs(2:3:end,1) = c1;
c1 = cell(1,1); c1{1} = 'HRR'; c1 = repmat(c1,10,1); HRs(3:3:end,1) = c1;

for i = 1 : 5
    o = strcat('O',num2str(i));
    c1 = cell(1,1); c1{1} = o; c1 = repmat(c1,3,1); OPs((i-1)*3+1:i*3,1) = c1;
end
OPs(16:end,1) = OPs(1:15,1);

% Create the within table
factorNames = {'HRs','VisualFeedback', 'ObstaclePos'};
within = table(HRs, VFs, OPs, 'VariableNames', factorNames);
```

```
% fit the repeated measures model
rm = fitrm(tbiases, 'V1-V30~1', 'WithinDesign', within);
[ranovatblb] = ranova(rm, 'WithinModel', 'HRs*VisualFeedback*ObstaclePos');
```

Please note the trick we used here; since we don't have a between subject variable we used 'V1-V30~1' instead of 'V1-V30~between variable'.

Now the most interesting part is how to run the post hoc analysis. The simplest one is to use the current within model that we have, running two-way interactions. We also, can specify what type of correction the program should use:

```
Mrm1 = multcompare(rm, 'HRs', 'By', 'VisualFeedback', 'ComparisonType', 'dunn-sidak');
Mrm2 = multcompare(rm, 'HRs', 'By', 'ObstaclePos', 'ComparisonType', 'bonferroni');
```

Now, the most challenging part is to do the 3-way comparison. To do so, we first need to transform our within design variables to categorical format. In addition, you need to combine, two of the variables and create a new variable:

```
within2 = within;
within2.HRs = categorical(within2.HRs);
within2.VisualFeedback = categorical(within2.VisualFeedback);
within2.ObstaclePos = categorical(within2.ObstaclePos);
within2.HRs_VisualFeedback = within2.HRs .* within2.VisualFeedback;

%%run my repeated measures anova for Directional Biases
rm = fitrm(tbiases, 'V1-V30~1', 'WithinDesign', within2); % overall fit
[ranovatblb] = ranova(rm, 'WithinModel', 'HRs*VisualFeedback*ObstaclePos');

Mrm1 = multcompare(rm, 'HRs', 'By', 'ObstaclePos');
Mrm2 = multcompare(rm, 'HRs_VisualFeedback', 'By', 'ObstaclePos');
```