

Modeling Notebook

1 md"# Modeling Notebook"

```
1 using CSV, DataFrames, Plots, Random, Statistics, GLM, StatsPlots, Tables, Printf  
    , MixedModels, Pipe, NaNStatistics
```

cond_num = 2

```
1 cond_num = 2
```

```
1 dfpre = CSV.File(raw"E:\+WORKSPACE\acro\affvids_physio_with_phobia.csv") |>  
DataFrames;
```

```
1 dfpre[:, :participant_num_str] = dfpre[:, :participant_num] .|> string
```

`mean_miss` (generic function with 1 method)

```
1 mean_miss(x) = mean(skipmissing(x))
```

std_miss (generic function with 1 method)

```
1 std_miss(x) = std(skipmissing(x))
```

```
df1pre =
```

	Column1	Unnamed: 0	video	video_condition	im_condition	pred_\
1	4	4	"social_low_1.m4v"	2	1	2
2	5	5	"social_high_6.mov"	2	2	1
3	6	6	"social_low_6.m4v"	2	1	1
4	11	11	"social_high_3.mov"	2	2	2
5	12	12	"social_low_3.m4v"	2	1	2
6	21	21	"social_low_2.mov"	2	1	2
7	22	22	"social_high_2.mov"	2	2	1
8	23	23	"social_high_1.mov"	2	2	1
9	24	24	"social_low_4.m4v"	2	1	2
10	33	33	"social_low_5.m4v"	2	1	2
: more						
1224	3785	3786	"social_low_2.mov"	2	1	1

```
1 df1pre = dfpre[dfpre[:, :video_condition] .== cond_num, :]
```

```
df2pre_pre =
```

	Column1	Unnamed: o	video	video_condition	im_condition	pred_
1	0	0	"spider_high_5.m4v"	3	2	2
2	1	1	"spider_low_3.m4v"	3	1	1
3	2	2	"spider_low_6.m4v"	3	1	1
4	3	3	"heights_low_3.mov"	1	1	2
5	4	4	"social_low_1.m4v"	2	1	2
6	5	5	"social_high_6.mov"	2	2	1
7	6	6	"social_low_6.m4v"	2	1	1
8	7	7	"spider_high_6.mov"	3	2	2
9	8	8	"heights_low_2.m4v"	1	1	2
10	9	9	"heights_high_1.m4v"	1	2	1
: more						
3691	3788	3789	"spider_low_5.mov"	3	1	1

```
1 df2pre_pre = @pipe dfpre |>
2   groupby(_, :participant_num) |>
3   transform(_, [:video_scr, :video_hp, :resp_arousal, :resp_fear,
4   :resp_valence] .=> mean_miss, [:video_scr, :video_hp, :resp_arousal,
5   :resp_fear, :resp_valence] .=> std_miss) |>
6   transform(_, [:video_scr, :video_scr_mean_miss] => ByRow(-) => :video_scr_cmc,
7   [:video_hp, :video_hp_mean_miss] => ByRow(-) => :video_hp_cmc,
8   [:resp_arousal, :resp_arousal_mean_miss] => ByRow(-) => :resp_arousal_cmc,
9   [:resp_fear, :resp_fear_mean_miss] => ByRow(-) => :resp_fear_cmc,
10  [:resp_valence, :resp_valence_mean_miss] => ByRow(-) => :resp_valence_cmc) |>
11  transform(_, [:video_scr_cmc, :video_scr_std_miss] => ByRow(/) =>
12    :video_scr_z, [:video_hp_cmc, :video_hp_std_miss] => ByRow(/) => :video_hp_z,
13    [:resp_arousal_cmc, :resp_arousal_std_miss] => ByRow(/) => :resp_arousal_z,
14    [:resp_fear_cmc, :resp_fear_std_miss] => ByRow(/) => :resp_fear_z,
15    [:resp_valence_cmc, :resp_valence_std_miss] => ByRow(/) => :resp_valence_z)
```

```
df2pre =
```

	Column1	Unnamed: 0	video	video_condition	im_condition	pred_\
1	4	4	"social_low_1.m4v"	2	1	2
2	5	5	"social_high_6.mov"	2	2	1
3	6	6	"social_low_6.m4v"	2	1	1
4	11	11	"social_high_3.mov"	2	2	2
5	12	12	"social_low_3.m4v"	2	1	2
6	21	21	"social_low_2.mov"	2	1	2
7	22	22	"social_high_2.mov"	2	2	1
8	23	23	"social_high_1.mov"	2	2	1
9	24	24	"social_low_4.m4v"	2	1	2
10	33	33	"social_low_5.m4v"	2	1	2
: more						
1224	3785	3786	"social_low_2.mov"	2	1	1

```
1 df2pre = df2pre_pre[df2pre_pre[:, :video_condition] .== cond_num, :]
```

Correlation Table

```
df3_extra =
```

	participant_num	social_phobia_mean_miss	video_scr_mean_miss	video_hp_mean_i
1	107	1.0	0.0476944	0.927861
2	108	1.66667	0.0416667	0.786829
3	109	1.75	0.0664167	0.863139
4	110	2.125	0.0737647	0.831937
5	112	1.70833	0.0319444	1.06551
6	114	1.83333	NaN	NaN
7	115	1.83333	0.0575833	0.813192
8	118	1.25	0.0490278	0.9876
9	119	1.91667	0.00288889	0.675694
10	120	1.70833	0.0414444	0.938516
: more				
105	232	2.125	0.0296364	0.992788

```
1 df3_extra = @pipe dfpre |>
2   groupby(_, :participant_num) |>
3   combine(_, [:social_phobia, :video_scr, :video_hp, :resp_arousal, :resp_fear,
:resp_valence] .=> mean_miss)
```

```
corcolnamesextra =
▶ ["social_phobia", "video_scr", "video_hp", "resp_arousal", "resp_fear", "resp_valence"
1 corcolnamesextra = names(df3_extra[:, 2:end]) .|> (x -> x[1:end-10])
```

```
cordfextra =
```

	social_phobia	video_scr	video_hp	resp_arousal	resp_fear	resp_valence
1	1.0	-0.129503	-0.0667437	0.0503779	0.333379	-0.26596
2	-0.129503	1.0	-0.183235	0.142326	0.0258892	-0.0571948
3	-0.0667437	-0.183235	1.0	0.0407244	-0.154579	0.141462
4	0.0503779	0.142326	0.0407244	1.0	0.378368	-0.0900944
5	0.333379	0.0258892	-0.154579	0.378368	1.0	-0.61302
6	-0.26596	-0.0571948	0.141462	-0.0900944	-0.61302	1.0

```
1 cordfextra = DataFrame(nancor(Matrix(df3_extra[,2:end])), corcolnamesextra)
```

Test Area

Column1	Unnamed: o	video	video_condition	im_condition	pred_\
1	4	"social_low_1.m4v"	2	1	2
2	5	"social_high_6.mov"	2	2	1
3	6	"social_low_6.m4v"	2	1	1
4	11	"social_high_3.mov"	2	2	2
5	12	"social_low_3.m4v"	2	1	2
6	21	"social_low_2.mov"	2	1	2
7	22	"social_high_2.mov"	2	2	1
8	23	"social_high_1.mov"	2	2	1
9	24	"social_low_4.m4v"	2	1	2
10	33	"social_low_5.m4v"	2	1	2
⋮ more					
1224	3785	3786	"social_low_2.mov"	2	1

```
1 df1pre
```

```
1 collist = ["video",
2 "participant_num",
3 "resp_exp_fear",
4 "rt_exp_fear",
5 "resp_current_anxiety",
6 "rt_current_anxiety",
7 "resp_fear",
8 "resp_anxiety",
9 "resp_arousal",
10 "resp_valence",
11 "rt_fear",
12 "rt_anxiety",
13 "rt_arousal",
14 "rt_valence",
15 "video_hp",
16 "video_scr",
17 "base_ECG",
18 "base_scr",
19 "hp_change_video",
20 "scr_change_video",
21 "social_phobia",
22 "participant_num_str"];
```

```
dfcleancol =
```

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_a
1	"social_low_1.m4v"	107	0.046	2.003	0.052
2	"social_high_6.mov"	107	0.061	2.103	0.059
3	"social_low_6.m4v"	107	0.042	2.336	0.046
4	"social_high_3.mov"	107	0.034	2.353	0.048
5	"social_low_3.m4v"	107	0.052	1.902	0.055
6	"social_low_2.mov"	107	0.039	3.721	0.059
7	"social_high_2.mov"	107	0.046	2.136	0.063
8	"social_high_1.mov"	107	0.051	1.201	0.038
9	"social_low_4.m4v"	107	0.068	1.318	0.059
10	"social_low_5.m4v"	107	0.053	2.203	0.052
: more					
1224	"social_low_2.mov"	232	0.216	4.872	0.115

```
1 dfcleancol = coalesce.(df1pre[ :, collist], 0)
```

```
1 collist_df2 = ["video",
2 "participant_num",
3 "resp_exp_fear",
4 "rt_exp_fear",
5 "resp_current_anxiety",
6 "rt_current_anxiety",
7 "resp_fear",
8 "resp_anxiety",
9 "resp_arousal",
10 "resp_valence",
11 "rt_fear",
12 "rt_anxiety",
13 "rt_arousal",
14 "rt_valence",
15 "video_hp",
16 "video_scr",
17 "base_ECG",
18 "base_scr",
19 "hp_change_video",
20 "scr_change_video",
21 "social_phobia",
22 "participant_num_str",
23 "video_scr_mean",
24 "resp_arousal_mean",
25 "resp_fear_mean",
26 "resp_valence_mean",
27 "video_scr_std",
28 "resp_arousal_std",
29 "resp_fear_std",
30 "resp_valence_std",
31 "video_scr_cmc",
32 "resp_arousal_cmc",
33 "resp_fear_cmc",
34 "resp_valence_cmc",
35 "video_scr_z",
36 "resp_arousal_z",
37 "resp_fear_z",
38 "resp_valence_z"];
```

```
ArgumentError: column name :video_scr_mean not found in the data frame
```

```
1. lookupname @ index.jl:413 [inlined]
2. getindex @ index.jl:422 [inlined]
3. getindex @ index.jl:423 [inlined]
4. (::DataFrames.var"#37#38"{DataFrames.Index})(::String) @ none:0
5. iterate @ generator.jl:47 [inlined]
6. collect_to! @ array.jl:845 [inlined]
7. collect_to_with_first! @ array.jl:823 [inlined]
8. collect(::Base.Generator{Vector{String}}, DataFrames.var"#37#38"
   {DataFrames.Index})) @ array.jl:797
9. getindex @ index.jl:428 [inlined]
10. getindex @ dataframe.jl:627 [inlined]
11. top-level scope @ Local: 1 [inlined]
```

```
1 df2cleancol = coalesce.(df2pre[:, collist_df2], 0)
```

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_anx
1	"social_low_5.m4v"	224	0.5	1.102	0.5
2	"social_high_5.m4v"	224	0.5	0.484	0.5
3	"social_low_2.mov"	224	0.5	0.801	0.5
4	"social_high_3.mov"	224	0.5	0.617	0.5
5	"social_low_6.m4v"	224	0.496	2.219	0.5
6	"social_high_1.mov"	224	0.498	3.054	0.5
7	"social_low_4.m4v"	224	0.5	1.635	0.5
8	"social_high_6.mov"	224	0.5	1.385	0.5
9	"social_low_1.m4v"	224	0.5	4.656	0.5
10	"social_low_3.m4v"	224	0.5	2.336	0.5
11	"social_high_2.mov"	224	0.472	1.335	0.5

```
1 dfcleancol[dfcleancol[:, :participant_num] .== 224, :]
```

```
▶ [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.103, 0.0, 0.0, 0.0, 0.0]
```

```
1 dfcleancol[:, "video_scr"]
```

```
comparevar (generic function with 1 method)
```

```
1 function comparevar(var1) #with phobia correlation
2     return cor(dfcleancol[:, "social_phobia"], dfcleancol[:, var1])
3 end
```

```
-0.08596314524418859
```

```
1 comparevar("video_scr")
```

```
0.08530891148907518
```

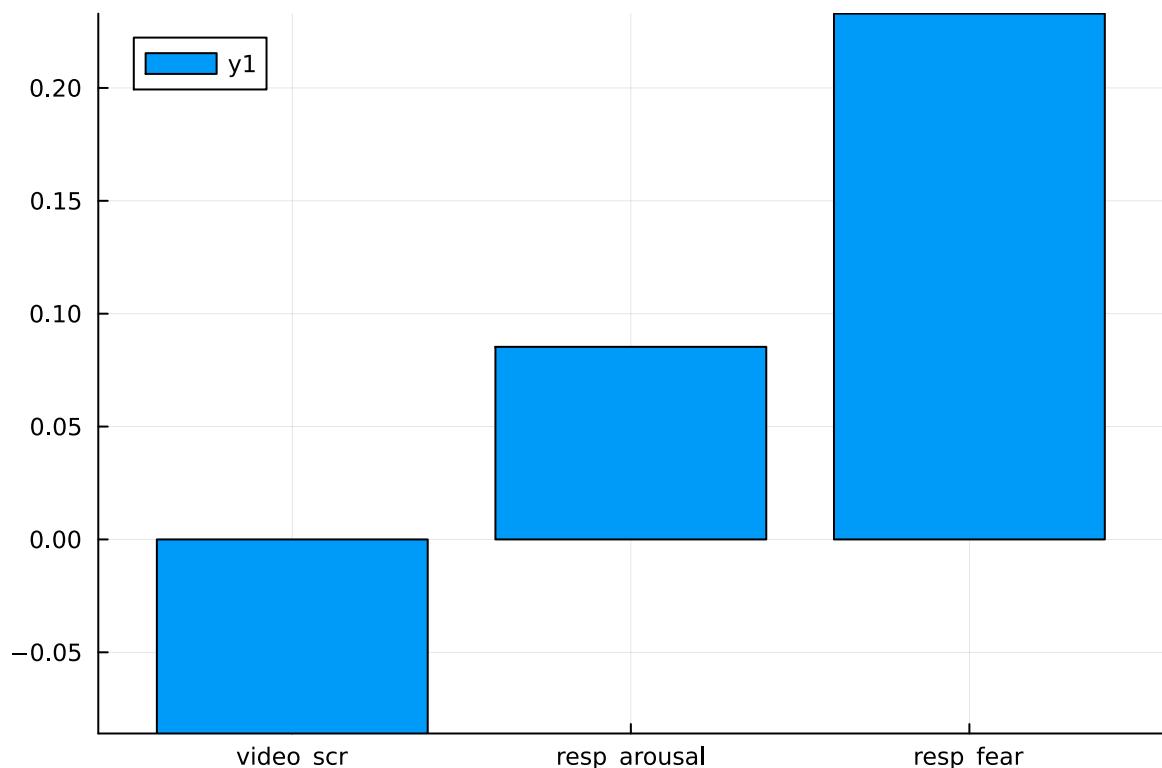
```
1 comparevar("resp_arousal")
```

```
0.23292830982908846
```

```
1 comparevar("resp_fear")
```

```
1 plotdata = [comparevar("video_scr"), comparevar("resp_arousal"),
comparevar("resp_fear")];
```

```
1 plotcols = ["video_scr", "resp_arousal", "resp_fear"];
```



```
1 plot(bar(plotcols, plotdata))
```

```
var1 = "video_scr"
```

```
1 var1 = "video_scr"
```

```
var2 = "resp_arousal"
```

```
1 var2 = "resp_arousal"
```

```
lm1 =  
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Vector{Float64}}}, GLM.DensePred2  
video_scr ~ 1 + resp_arousal
```

Coefficients:

	Coef.	Std. Error	t	Pr(> t)	Lower 95%	Upper 95%
(Intercept)	0.0278991	0.00224479	12.43	<1e-32	0.023495	0.0323032
resp_arousal	0.00286125	0.00404107	0.71	0.4791	-0.00506695	0.0107895

```
1 lm1 = lm(@formula(video_scr~ resp_arousal), dfcleancol)
```

0.027899110461286954

```
1 coef(lm1)[1]
```

```
gdf =
```

GroupedDataFrame with 105 groups based on key: participant_num_str

First Group (12 rows): participant_num_str = "107"

Row	video	participant_num	resp_exp_fear	rt_exp_fear	re:
	String31	Int64	Float64	Float64	Fl
1	social_low_1.m4v	107	0.046	2.003	
2	social_high_6.mov	107	0.061	2.103	
3	social_low_6.m4v	107	0.042	2.336	
4	social_high_3.mov	107	0.034	2.353	
5	social_low_3.m4v	107	0.052	1.902	
6	social_low_2.mov	107	0.039	3.721	
7	social_high_2.mov	107	0.046	2.136	
8	social_high_1.mov	107	0.051	1.201	
9	social_low_4.m4v	107	0.068	1.318	
10	social_low_5.m4v	107	0.053	2.203	
11	social_high_4_replacement.mov	107	0.077	1.168	
12	social_high_5.m4v	107	0.034	1.585	

:

Last Group (12 rows): participant_num_str = "232"

Row	video	participant_num	resp_exp_fear	rt_exp_fear	re:
	String31	Int64	Float64	Float64	Fl
1	social_low_6.m4v	232	0.081	4.005	
2	social_high_4_replacement.mov	232	0.069	5.773	
3	social_high_5.m4v	232	0.163	2.903	
4	social_low_5.m4v	232	0.285	73.148	
5	social_high_6.mov	232	0.18	6.474	
6	social_low_3.m4v	232	0.09	4.038	
7	social_high_2.mov	232	0.227	3.354	
8	social_low_4.m4v	232	0.18	3.103	
9	social_high_1.mov	232	0.185	7.392	
10	social_low_1.m4v	232	0.221	9.177	
11	social_high_3.mov	232	0.192	10.862	
12	social_low_2.mov	232	0.216	4.872	

```
1 gdf = groupby(dfcleancol, :participant_num_str)
```

	video	participant_num	resp_exp_fear	rt_exp_fear	resp
1	"social_high_4_replacement.mov"	114	0.541	2.803	0.5
2	"social_low_3.m4v"	114	0.582	2.319	0.6
3	"social_high_1.mov"	114	0.5	2.002	0.48
4	"social_low_2.mov"	114	0.5	1.986	0.5
5	"social_high_3.mov"	114	0.605	1.268	0.61
6	"social_low_4.m4v"	114	0.703	2.57	0.71
7	"social_high_6.mov"	114	0.448	3.153	0.3
8	"social_low_1.m4v"	114	0.568	2.67	0.51
9	"social_low_5.m4v"	114	0.36	2.053	0.3
10	"social_low_6.m4v"	114	0.14	3.087	0.1
11	"social_high_2.mov"	114	0.413	3.054	0.31
12	"social_high_5.m4v"	114	0.332	2.453	0.21

```
1 gdf[6]
```

```
1 collist2 = [
2 "resp_exp_fear",
3 "rt_exp_fear",
4 "resp_current_anxiety",
5 "rt_current_anxiety",
6 "resp_fear",
7 "resp_anxiety",
8 "resp_arousal",
9 "resp_valence",
10 "rt_fear",
11 "rt_anxiety",
12 "rt_arousal",
13 "rt_valence",
14 "video_hp",
15 "video_scr",
16 "base_ECG",
17 "base_scr",
18 "hp_change_video",
19 "scr_change_video",
20 "social_phobia"];
```

```

meantest (generic function with 1 method)
1 function meantest(series1, series2)
2   if sum(series1) == 0 && mean(series1) == 0
3     return missing
4   end
5   dfloc = DataFrame(a=series1, b=series2)
6   lm1 = lm(@formula(a~ b),dfloc)
7   return (coef(lm1)[1], coef(lm1)[2], loglikelihood(lm1))
8 end

```

Fitting SCR to Arousal

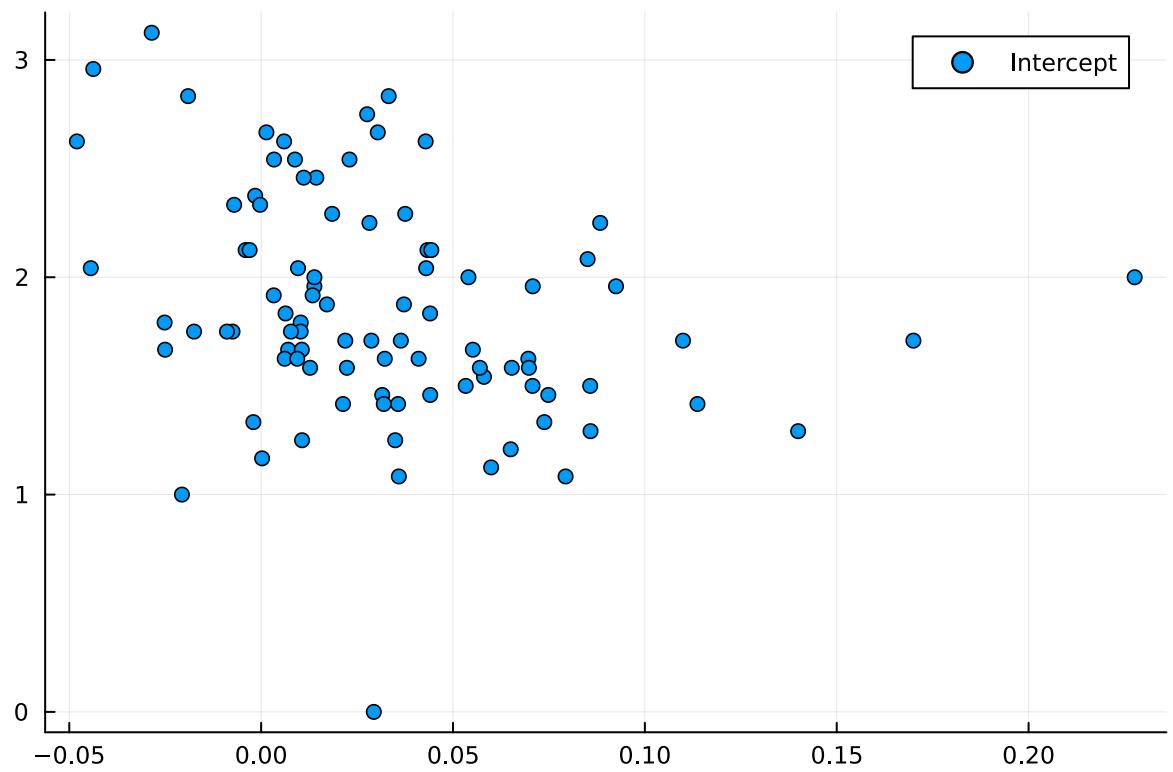
`df2_subjlvl =`

	participant_num_str	video_scr_resp_arousal	social_phobia
1	"107"	► (-0.0206453, 0.0379001, 36.2333)	1.0
2	"108"	► (0.00703823, 0.0203924, 24.5381)	1.66667
3	"109"	► (-0.0174806, 0.105126, 20.3145)	1.75
4	"110"	► (0.0433452, 0.0452475, 18.423)	2.125
5	"112"	► (0.0219211, 0.0170451, 23.5583)	1.70833
6	"115"	► (0.0440328, 0.0163736, 22.6576)	1.83333
7	"118"	► (0.0349378, -0.00142819, 27.5753)	1.25
8	"120"	► (0.0364008, -0.0270166, 24.247)	1.70833
9	"122"	► (-0.00202442, 0.0425042, 28.896)	1.33333
10	"123"	► (0.00598515, 0.0616688, 20.2089)	2.625
⋮ more			
92	"232"	► (-0.00300193, 0.063561, 29.016)	2.125

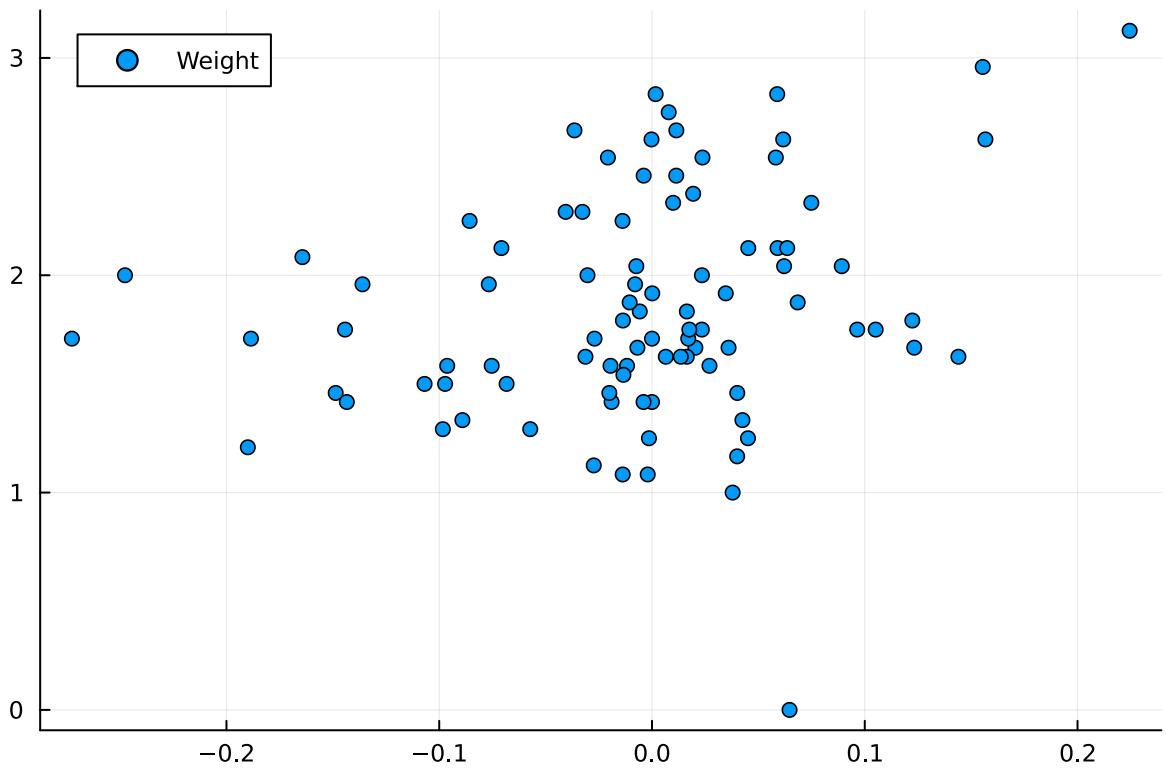
```

1 df2_subjlvl = combine(gdf, [:video_scr, :resp_arousal] => meantest,
:social_phobia => mean, renamecols=false) |> dropmissing

```



```
1 scatter(df2_subjlvl.video_scr_resp_arousal .|> x -> x[1],  
df2_subjlvl.social_phobia, label="Intercept")
```



```
1 scatter(df2_subjlvl.video_scr_resp_arousal .|> x -> x[2],
df2_subjlvl.social_phobia, label="Weight")
```

`scr_to_aro_int = -0.29192217590311786`

```
1 scr_to_aro_int = cor(df2_subjlvl.video_scr_resp_arousal .|> x -> x[1],
df2_subjlvl.social_phobia)
```

`scr_to_aro_coef = 0.24817338111990372`

```
1 scr_to_aro_coef = cor(df2_subjlvl.video_scr_resp_arousal .|> x -> x[2],
df2_subjlvl.social_phobia)
```

▶ [36.2333, 24.5381, 20.3145, 18.423, 23.5583, 22.6576, 27.5753, 24.247, 28.896, 20.2089]

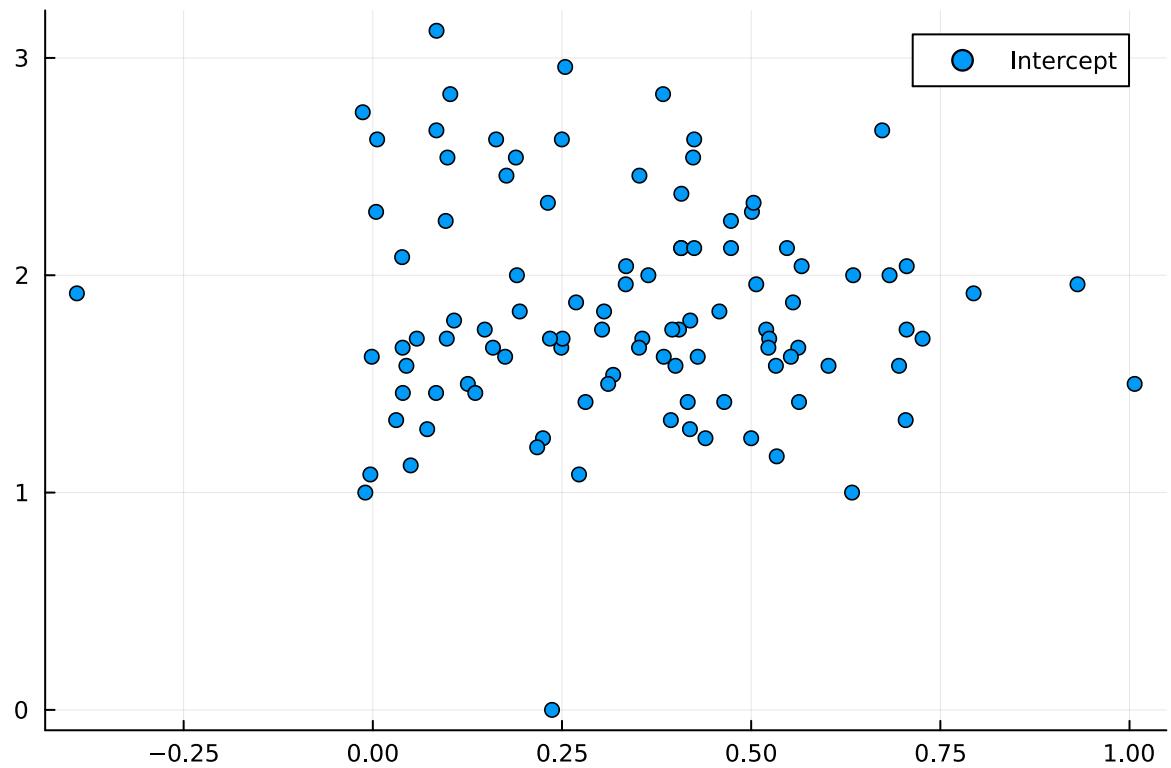
```
1 df2_subjlvl.video_scr_resp_arousal .|> x -> x[3]
```

Fitting Arousal to Fear Response

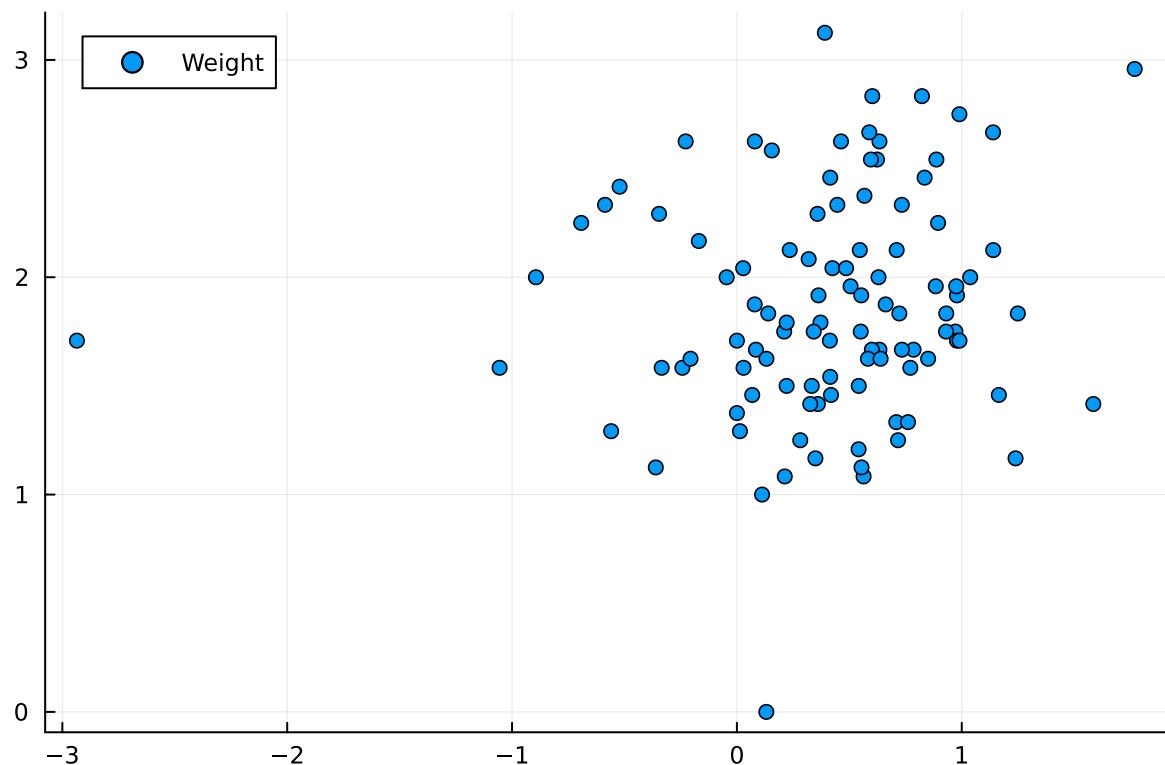
```
df2_subjlvl_2 =
```

	participant_num_str	resp_arousal_resp_fear	social_phobia
1	"107"	► (0.633317, 0.111973, 3.04524)	1.0
2	"108"	► (0.249079, 0.633906, 0.911873)	1.66667
3	"109"	► (0.519882, 0.971804, -1.52019)	1.75
4	"110"	► (0.47336, 0.546476, 0.58467)	2.125
5	"112"	► (0.0581016, 0.978455, 12.6447)	1.70833
6	"114"	► (0.30558, 0.93106, 1.35341)	1.83333
7	"115"	► (0.439603, 0.139271, 9.79565)	1.83333
8	"118"	► (0.356173, 0.716763, 3.05325)	1.25
9	"119"	► (0.0308601, 0.978662, 9.47965)	1.91667
10	"120"	► (0.24983, -2.93535, -3.72305)	1.70833
⋮ more			
104	"232"	► (0.158778, 1.13998, 6.79258)	2.125

```
1 df2_subjlvl_2 = combine(gdf, [:resp_arousal, :resp_fear] => meantest,  
:social_phobia => mean, renamecols=false) |> dropmissing
```



```
1 scatter(df2_subjlvl_2.resp_arousal_resp_fear .|> x -> x[1],  
df2_subjlvl.social_phobia, label="Intercept")
```



```
1 scatter(df2_subjlv2.resp_arousal_resp_fear .|> x -> x[2],
df2_subjlv2.social_phobia, label="Weight")
```

```
aro_to_fear_int = -0.11692501229616001
```

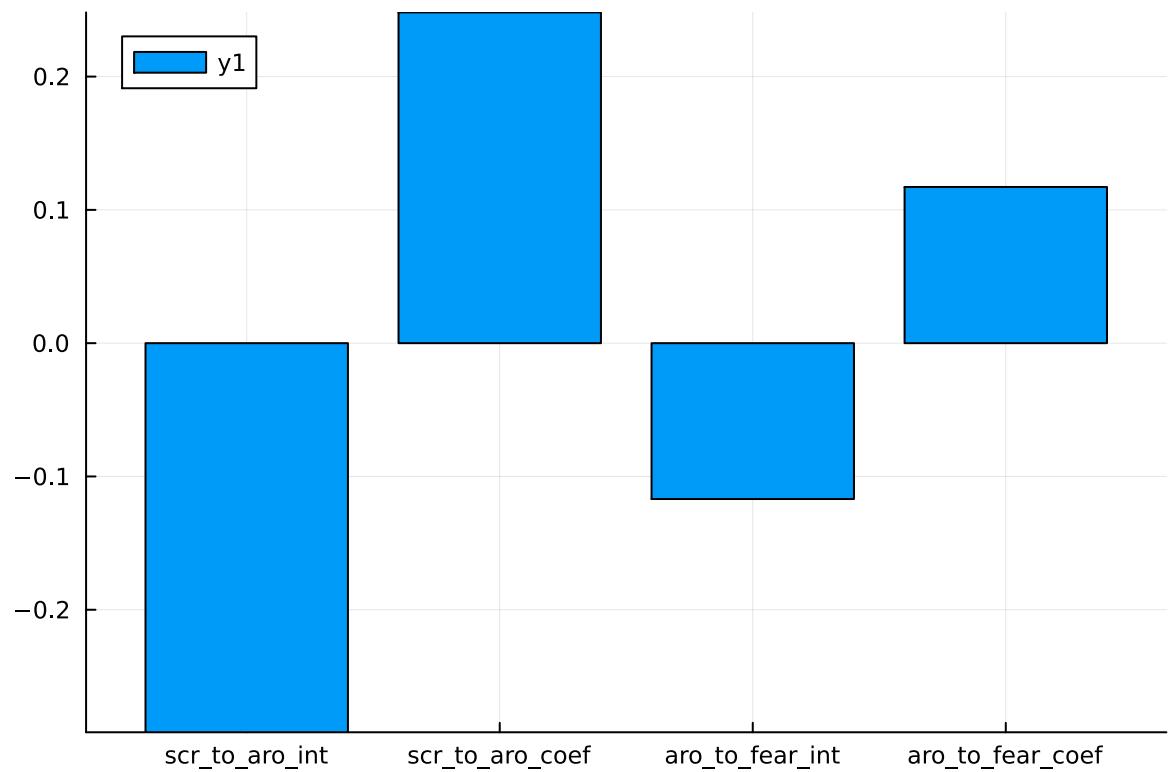
```
1 aro_to_fear_int = cor(df2_subjlv2.resp_arousal_resp_fear .|> x -> x[1],
df2_subjlv2.social_phobia)
```

```
aro_to_fear_coef = 0.11718565225602073
```

```
1 aro_to_fear_coef = cor(df2_subjlv2.resp_arousal_resp_fear .|> x -> x[2],
df2_subjlv2.social_phobia)
```

```
1 plotdata2 = [scr_to_aro_int, scr_to_aro_coef, aro_to_fear_int, aro_to_fear_coef];
```

```
1 plotcols2 = ["scr_to_aro_int", "scr_to_aro_coef", "aro_to_fear_int",
"aro_to_fear_coef"];
```



```
1 plot(bar(plotcols2, plotdata2))
```

Test Area 2

```
1 df2_subjlvl_hp = combine(gdf, :social_phobia => mean, renamecols=false);
```

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_a
1	"social_low_1.m4v"	107	0.046	2.003	0.052
2	"social_high_6.mov"	107	0.061	2.103	0.059
3	"social_low_6.m4v"	107	0.042	2.336	0.046
4	"social_high_3.mov"	107	0.034	2.353	0.048
5	"social_low_3.m4v"	107	0.052	1.902	0.055
6	"social_low_2.mov"	107	0.039	3.721	0.059
7	"social_high_2.mov"	107	0.046	2.136	0.063
8	"social_high_1.mov"	107	0.051	1.201	0.038
9	"social_low_4.m4v"	107	0.068	1.318	0.059
10	"social_low_5.m4v"	107	0.053	2.203	0.052
: more					
1224	"social_low_2.mov"	232	0.216	4.872	0.115

1 dfcleancol

mm2 =

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.4660	0.0183	25.46	<1e-99	0.1641
video_scr	0.1511	0.1887	0.80	0.4232	
Residual	0.2422				

```
Minimizing 2      Time: 0:00:00 (74.00 ms/it)
objective: 289.7011283734115
```

```
Minimizing 25      Time: 0:00:00 ( 9.17 ms/it)
```

	Column	Variance	Std.Dev
participant_num_str	(Intercept)	0.026937	0.164125
Residual		0.058646	0.242170

```
1 VarCorr(mm2)
```

```
▶ [1×105 Matrix{Float64}:
 0.159118  0.0133825  0.125942  0.193821  ...
  ...  0.0921158  0.0926508  -0.129192]
```

```
1 ranef(mm2)
```

0.02288805656593195

```
1 var(only(ranef(mm2)))
```

	participant_num_str	(Intercept)
1	"107"	0.159118
2	"108"	0.0133825
3	"109"	0.125942
4	"110"	0.193821
5	"112"	0.0358879
6	"114"	0.130873
7	"115"	0.00643649
8	"118"	-0.0406189
9	"119"	-0.173208
10	"120"	-0.203024
:	more	
105	"232"	-0.129192

```
1 DataFrame(only(raneftables(mm2)))
```

Mixed Models

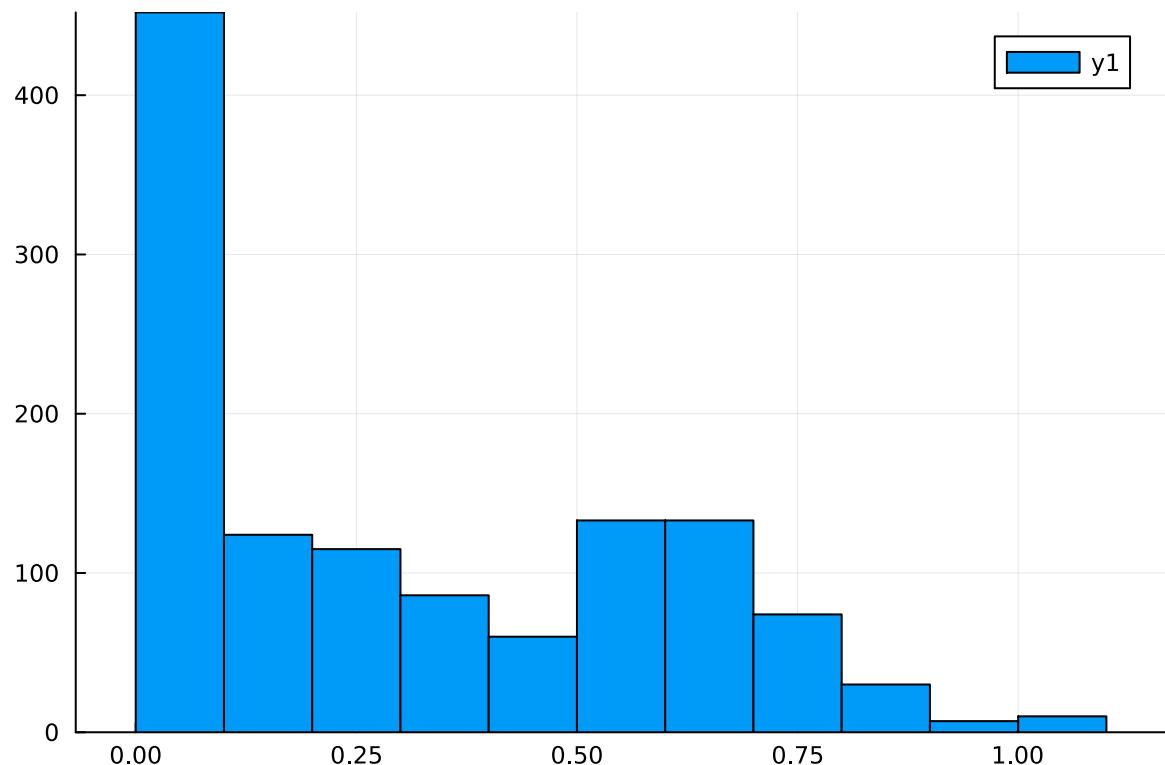
```
1 md"## Mixed Models"
```

```
dfcleancol2 =
```

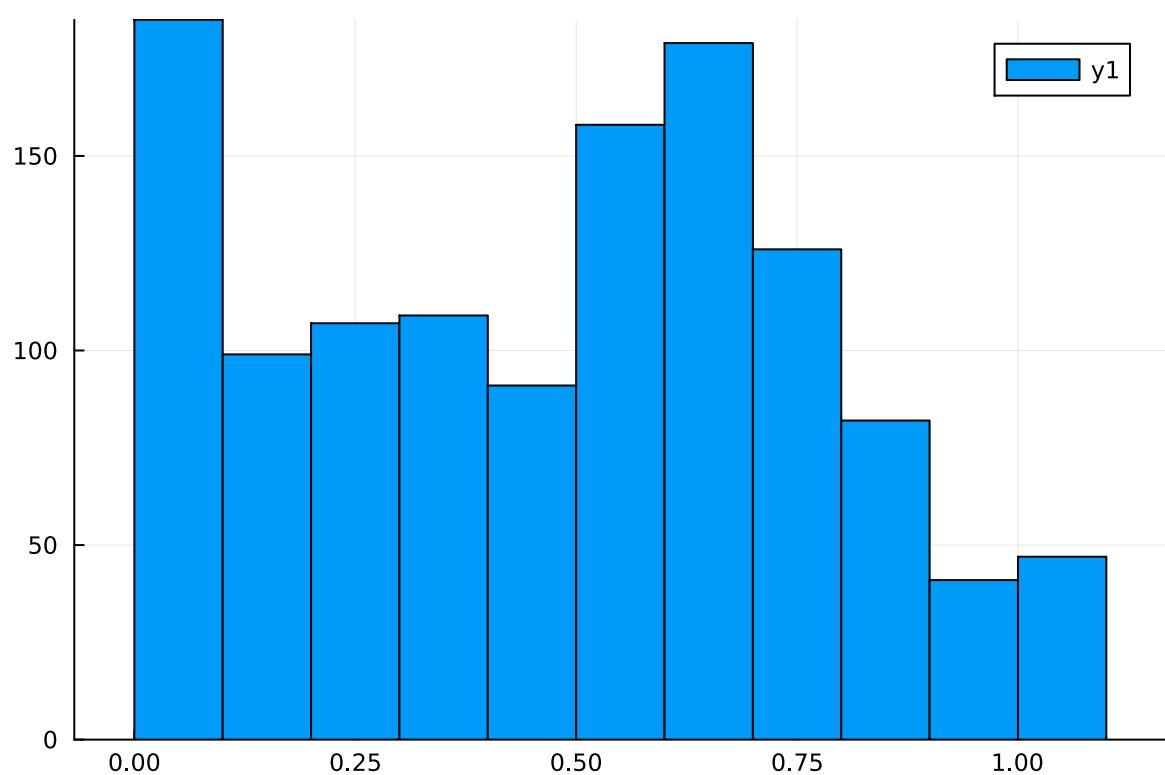
	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_a
1	"social_low_1.m4v"	107	0.046	2.003	0.052
2	"social_high_6.mov"	107	0.061	2.103	0.059
3	"social_low_6.m4v"	107	0.042	2.336	0.046
4	"social_high_3.mov"	107	0.034	2.353	0.048
5	"social_low_3.m4v"	107	0.052	1.902	0.055
6	"social_low_2.mov"	107	0.039	3.721	0.059
7	"social_high_2.mov"	107	0.046	2.136	0.063
8	"social_high_1.mov"	107	0.051	1.201	0.038
9	"social_low_4.m4v"	107	0.068	1.318	0.059
10	"social_low_5.m4v"	107	0.053	2.203	0.052
: more					
1224	"social_low_2.mov"	232	0.216	4.872	0.115

```
1 dfcleancol2 = @pipe dfcleancol |>
2     groupby(_, :participant_num_str) |>
3     transform(_, [:video_scr, :video_hp, :resp_arousal,
4      :scr_change_video, :resp_valence] .=> mean)|>
5     transform(_, [:video_scr, :video_scr_mean] => ByRow(-) =>
6      :video_scr_cmc,
7      [:video_hp, :video_hp_mean] => ByRow(-) => :video_hp_cmc,
8      [:resp_arousal, :resp_arousal_mean] => ByRow(-) =>
9      :resp_arousal_cmc,
10     [:scr_change_video, :scr_change_video_mean] => ByRow(-) =>
11     :scr_change_video_cmc,
12     [:resp_valence, :resp_valence_mean] => ByRow(-) =>
13     :resp_valence_cmc)
```

```
get_trunc (generic function with 1 method)
```



```
1 histogram(dfcleancol2.resp_fear)
```



```
1 histogram(dfcleancol2.resp_arousal)
```

Model 1: src to arousal

```
mm3 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.4623	0.0294	15.71	<1e-54	0.1644
video_scr_cmc	0.0992	0.2068	0.48	0.6314	0.6593
video_scr_mean	0.2780	0.8139	0.34	0.7327	
Residual	0.2408				

```
1 mm3 = fit(LinearMixedModel, @formula(resp_arousal ~ video_scr_cmc +  
video_scr_mean + (video_scr_cmc|participant_num_str)), dfcleancol2)
```

Scr to arousal, loglike: -92.61125712525266

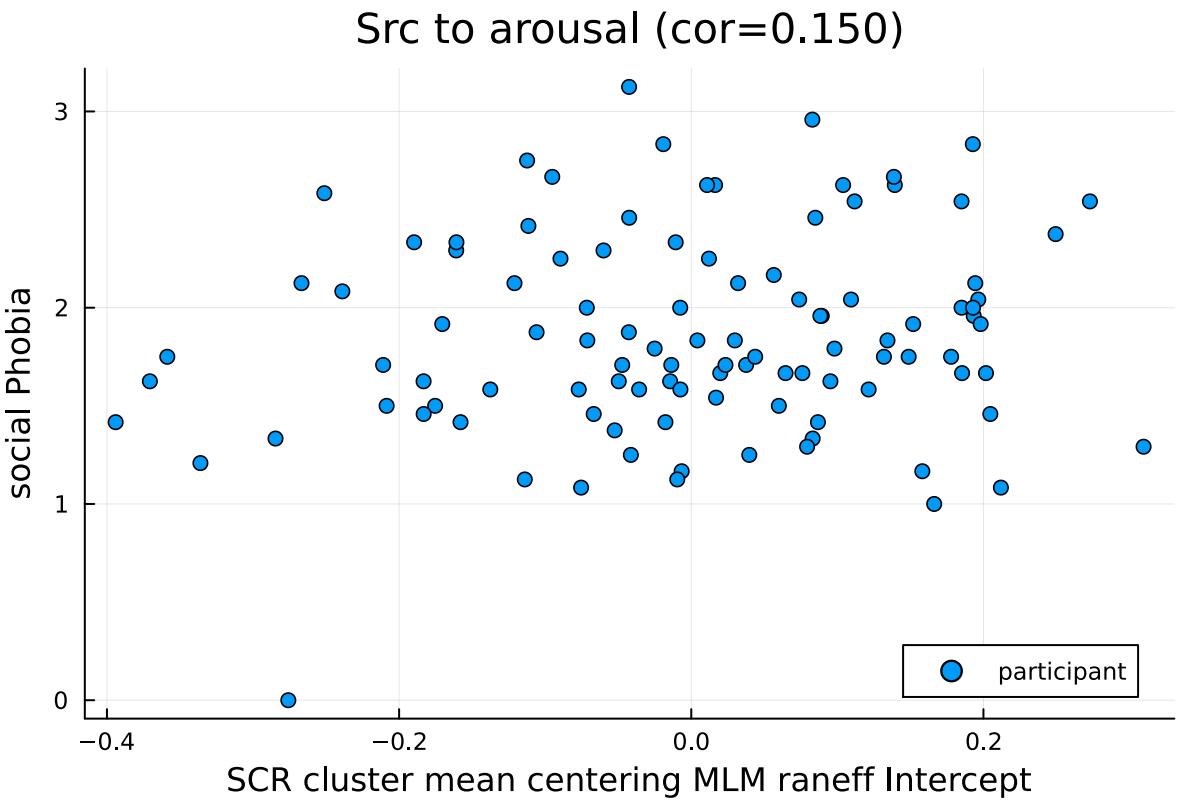
```
1 md"#### Scr to arousal, loglike: $(loglikelihood(mm3))"
```

Scr to arousal, AIC: 199.22251425050533

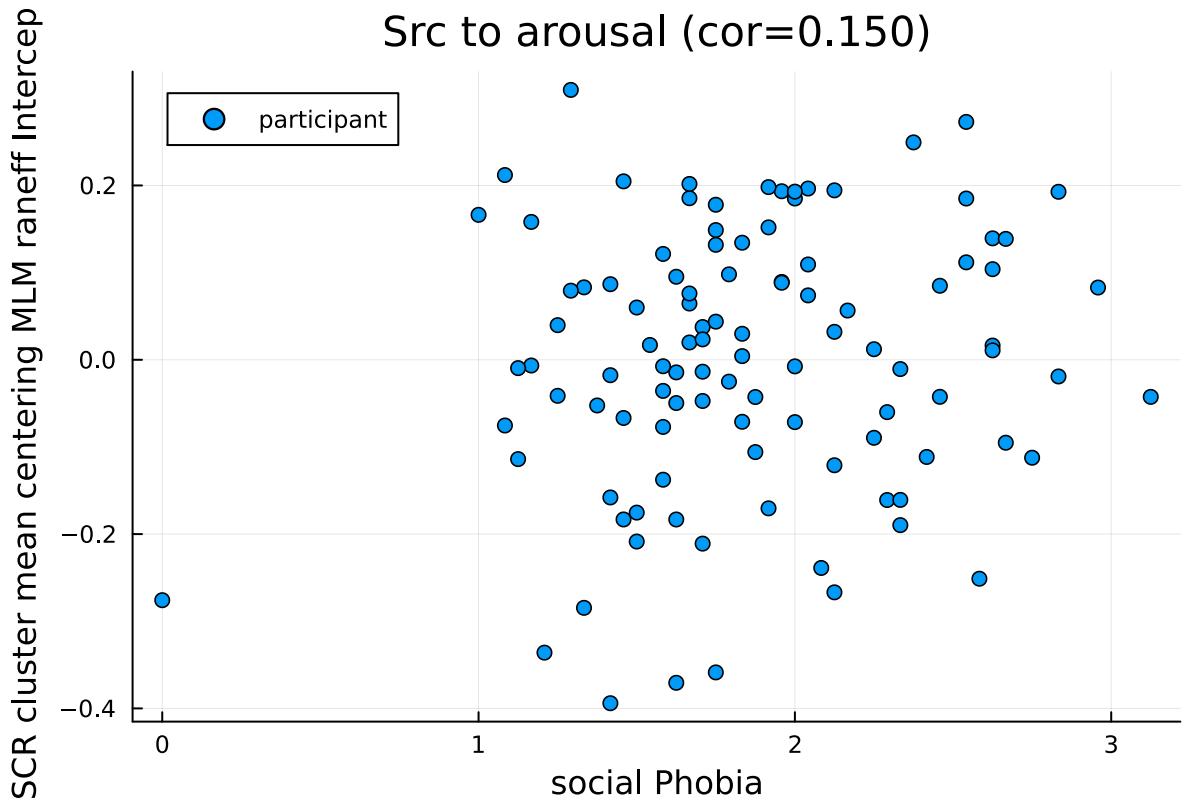
```
1 md"#### Scr to arousal, AIC: $(aic(mm3))"
```

```
1 mm3coef_i = cor(ranef(mm3)[1][1,:], df2_subjlvl_hp.social_phobia);
```

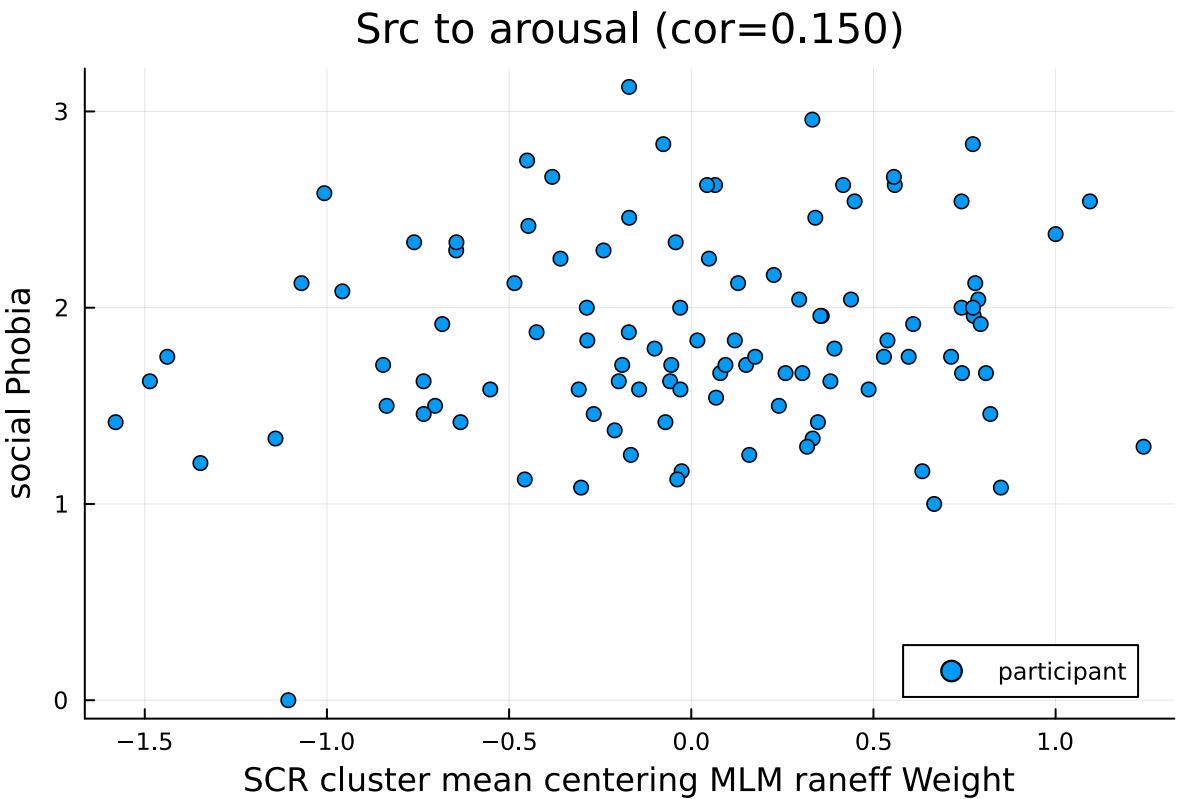
```
1 mm3coef_w = cor(ranef(mm3)[1][2,:], df2_subjlvl_hp.social_phobia);
```



```
1 scatter(ranef(mm3)[1][1,:], df2_subjlvl_hp.social_phobia, label="participant",
  title="Src to arousal (cor=$(get_trunc(mm3coef_i)))", xlabel="SCR cluster mean
  centering MLM raneff Intercept", ylabel="social Phobia")
```



```
1 scatter( df2_subjlv_hp.social_phobia, ranef(mm3)[1][1,:], label="participant",
  title="Src to arousal (cor=$(get_trunc(mm3coef_i)))", ylabel="SCR cluster mean
  centering MLM raneff Intercept", xlabel="social Phobia")
```



```
1 scatter(ranef(mm3)[1][2,:], df2_subjlvl_hp.social_phobia, label="participant",
  title="Src to arousal (cor=$(get_trunc(mm3$coef_w)))", xlabel="SCR cluster mean
  centering MLM raneff Weight", ylabel="social Phobia")
```

Model 2: arousal to fear

```
1 md## Model 2: arousal to fear"
```

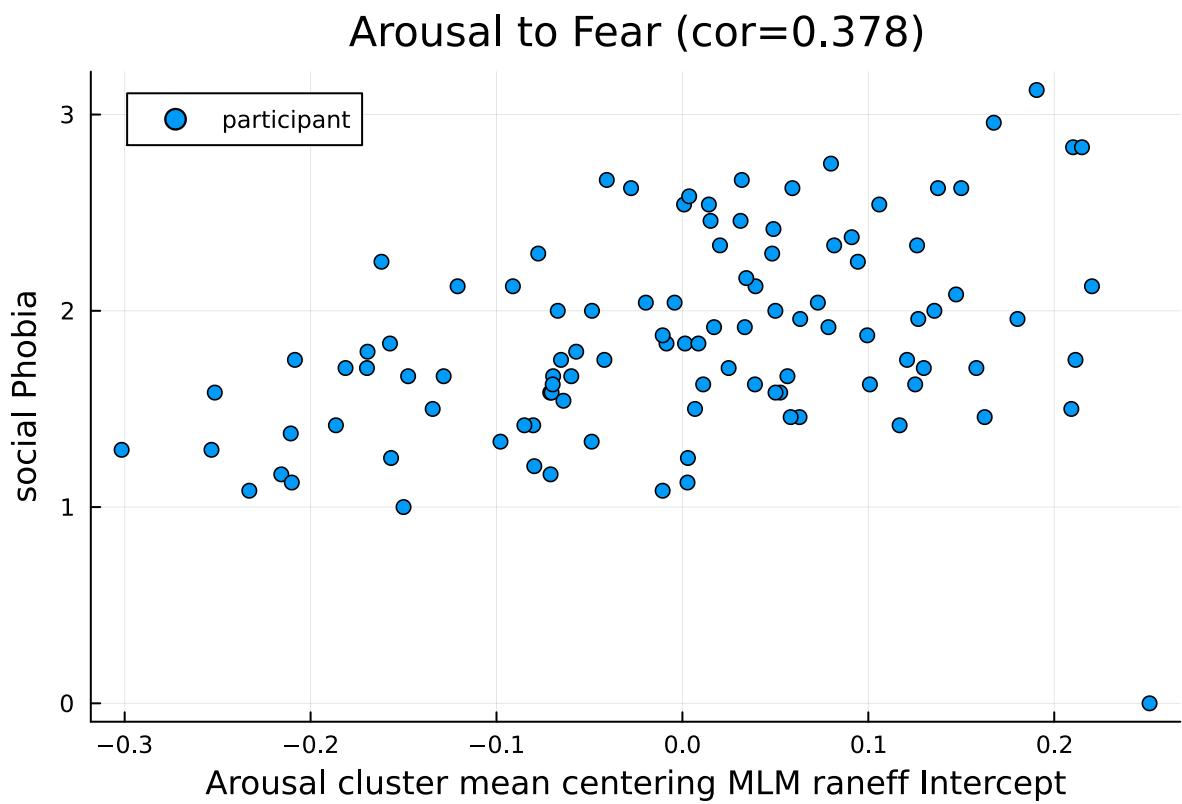
```
mm4 =
```

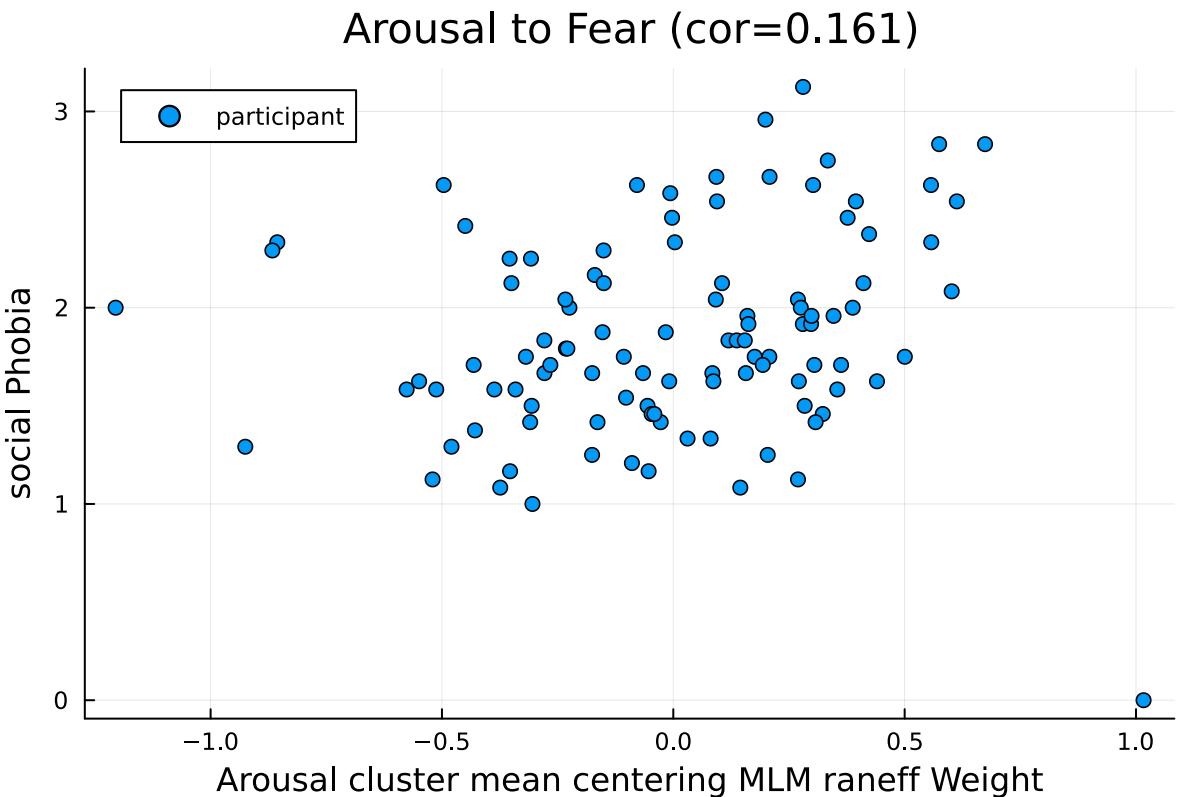
	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.1348	0.0367	3.67	0.0002	0.1327
resp_arousal_cmc	0.4453	0.0519	8.58	<1e-17	0.4377
resp_arousal_mean	0.3552	0.0720	4.93	<1e-06	
Residual	0.1964				

```
1 mm4 = fit(LinearMixedModel, @formula(resp_fear ~ resp_arousal_cmc +
  resp_arousal_mean + (resp_arousal_cmc|participant_num_str)), dfcleancol2)
```

Arousal to fear, loglike: 104.1217829473016

Arousal to fear, AIC: -194.2435658946032





```
1 scatter(ranef(mm4)[1][2,:], df2_subjlvl_hp.social_phobia, label="participant",
  title="Arousal to Fear (cor=$(get_trunc(mm4coef_w)))", xlabel="Arousal cluster
  mean centering MLM raneff Weight", ylabel="social Phobia")
```

Model 3: scr to fear

```
1 md## Model 3: scr to fear"
```

```
mm6 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.3147	0.0261	12.07	<1e-32	0.1391
video_scr_cmc	0.2335	0.1978	1.18	0.2377	0.3732
video_scr_mean	-0.4365	0.7267	-0.60	0.5481	
Residual	0.2414				

```
1 mm6 = fit(LinearMixedModel, @formula(resp_fear ~ video_scr_cmc + video_scr_mean
  + (video_scr_cmc|participant_num_str)), dfcleancol2)
```

Scr to fear, loglike: -80.41985151658388

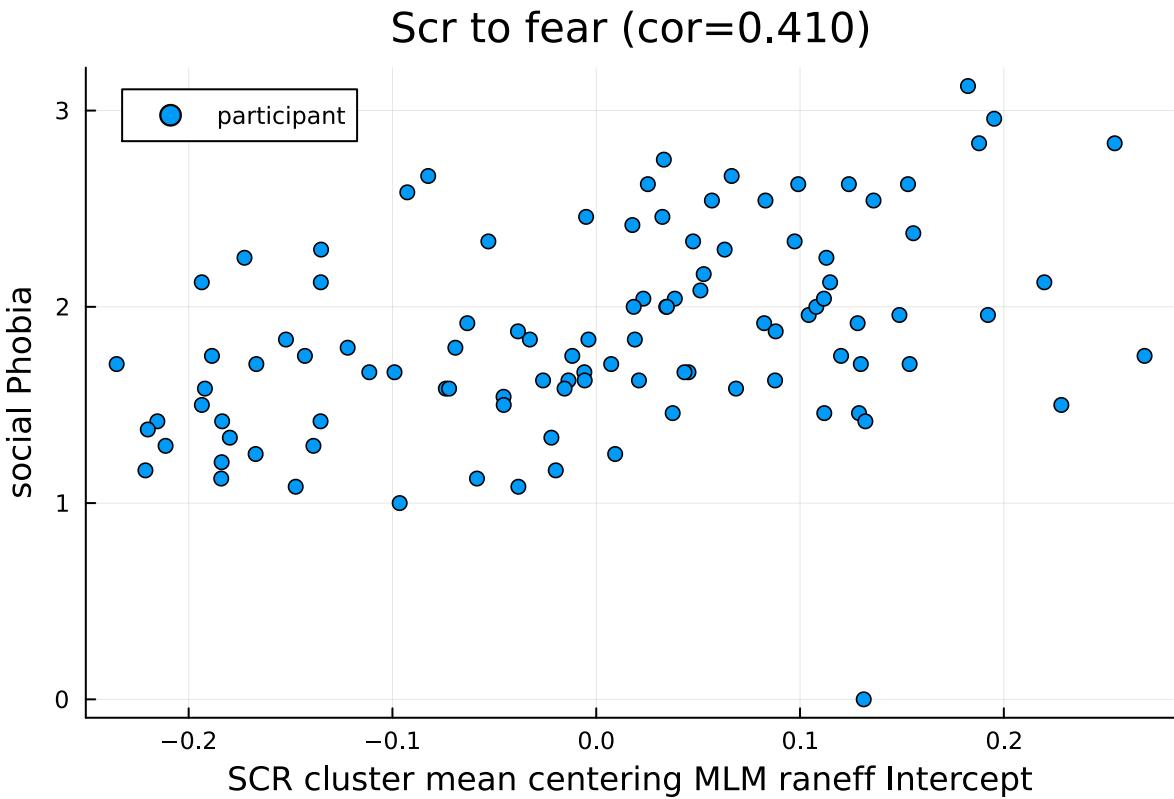
```
1 md#### Scr to fear, loglike: $(loglikelihood(mm6))"
```

Scr to fear, AIC: 174.83970303316775

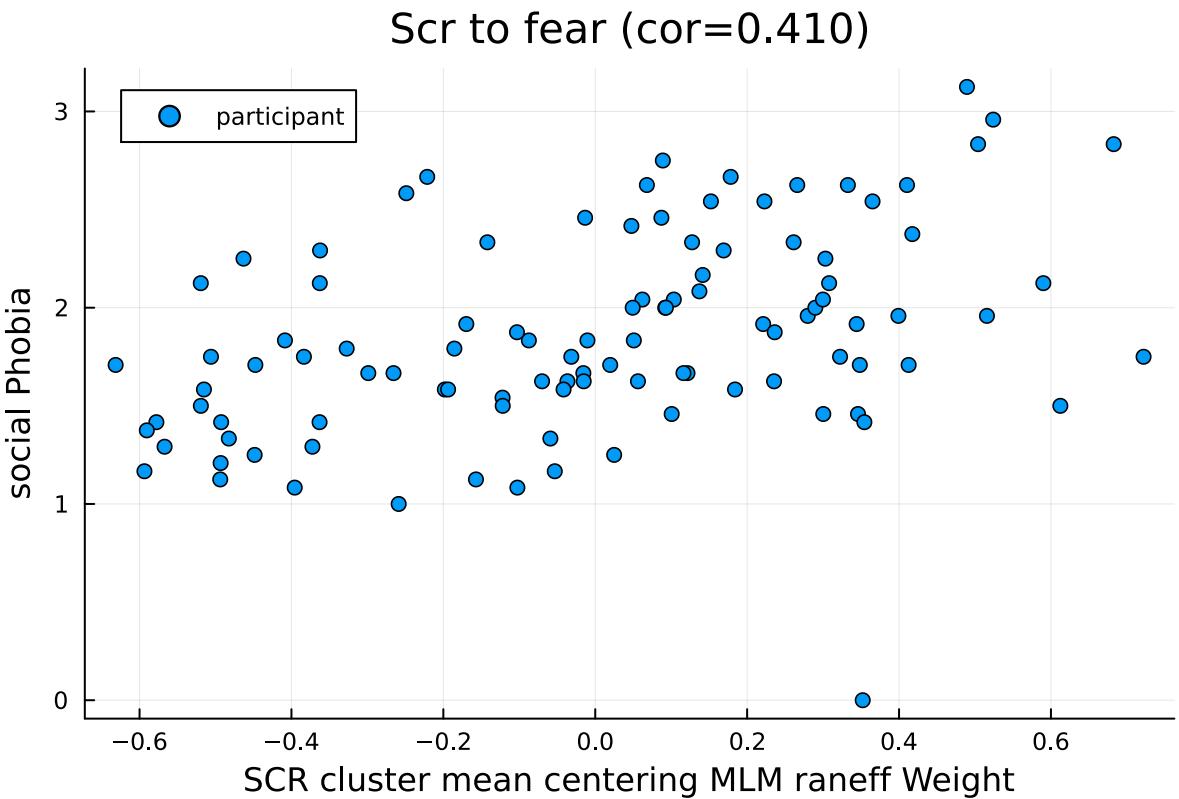
```
1 md"#### Scr to fear, AIC: $(aic(mm6))"
```

```
1 mm6coef_i = cor(ranef(mm6)[1][1,:], df2_subjlvl_hp.social_phobia);
```

```
1 mm6coef_w = cor(ranef(mm6)[1][2,:], df2_subjlvl_hp.social_phobia);
```



```
1 scatter(ranef(mm6)[1][1,:], df2_subjlvl_hp.social_phobia, label="participant",
  title="Scr to fear (cor=$(get_trunc(mm6coef_i)))", xlabel="SCR cluster mean
  centering MLM raneff Intercept", ylabel="social Phobia")
```



```
1 scatter(ranef(mm6)[1][2,:], df2_subjlv_hp.social_phobia, label="participant",
  title="Scr to fear (cor=$(get_trunc(mm6$coef_w)))", xlabel="SCR cluster mean
  centering MLM raneff Weight", ylabel="social Phobia")
```

Model 4: scr to valence

```
1 md"## Model 4: scr to valence"
```

```
mm12 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.4842	0.0191	25.33	<1e-99	0.0780
video_scr_cmc	-0.5671	0.2247	-2.52	0.0116	0.1212
video_scr_mean	1.2473	0.5340	2.34	0.0195	
Residual	0.2804				

```
1 mm12 = fit(LinearMixedModel, @formula(resp_valence ~ video_scr_cmc +
  video_scr_mean + (video_scr_cmc|participant_num_str)), dfcleancl2)
```

Scr to arousal, loglike: -214.36744614123768

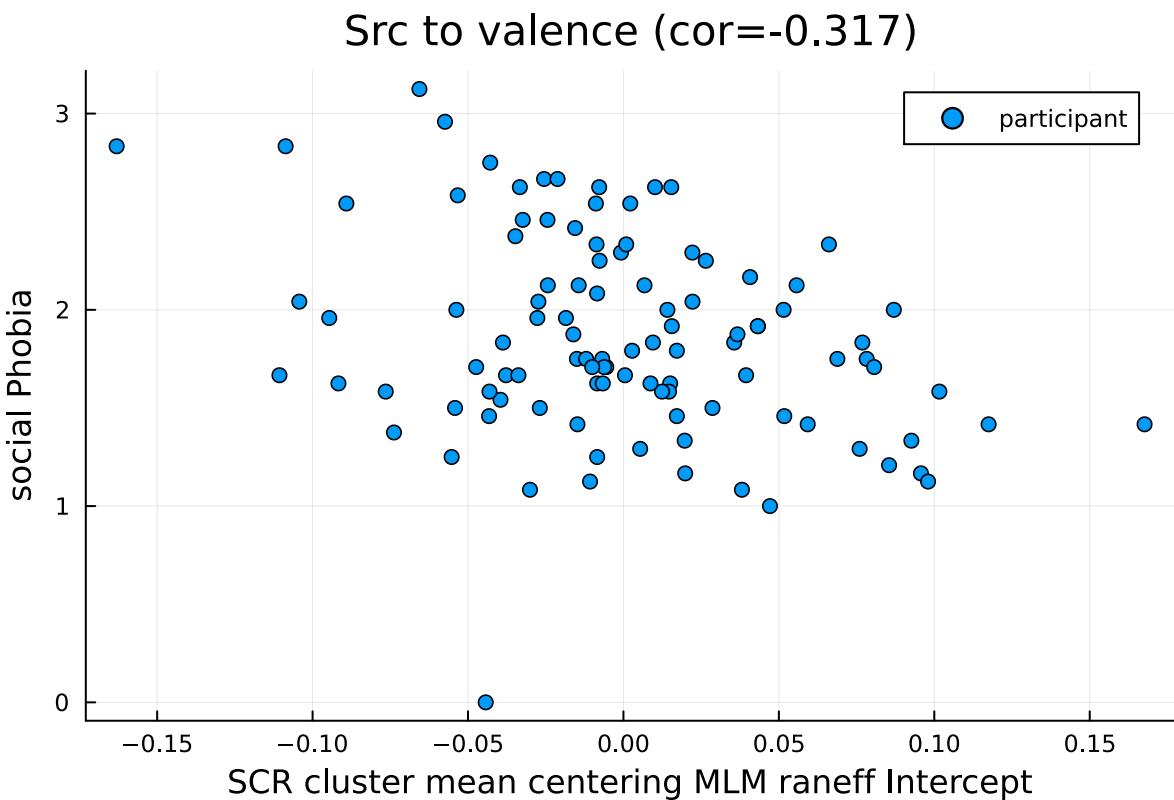
```
1 md"#### Scr to arousal, loglike: $(loglikelihood(mm12))"
```

Scr to arousal, AIC: 442.73489228247536

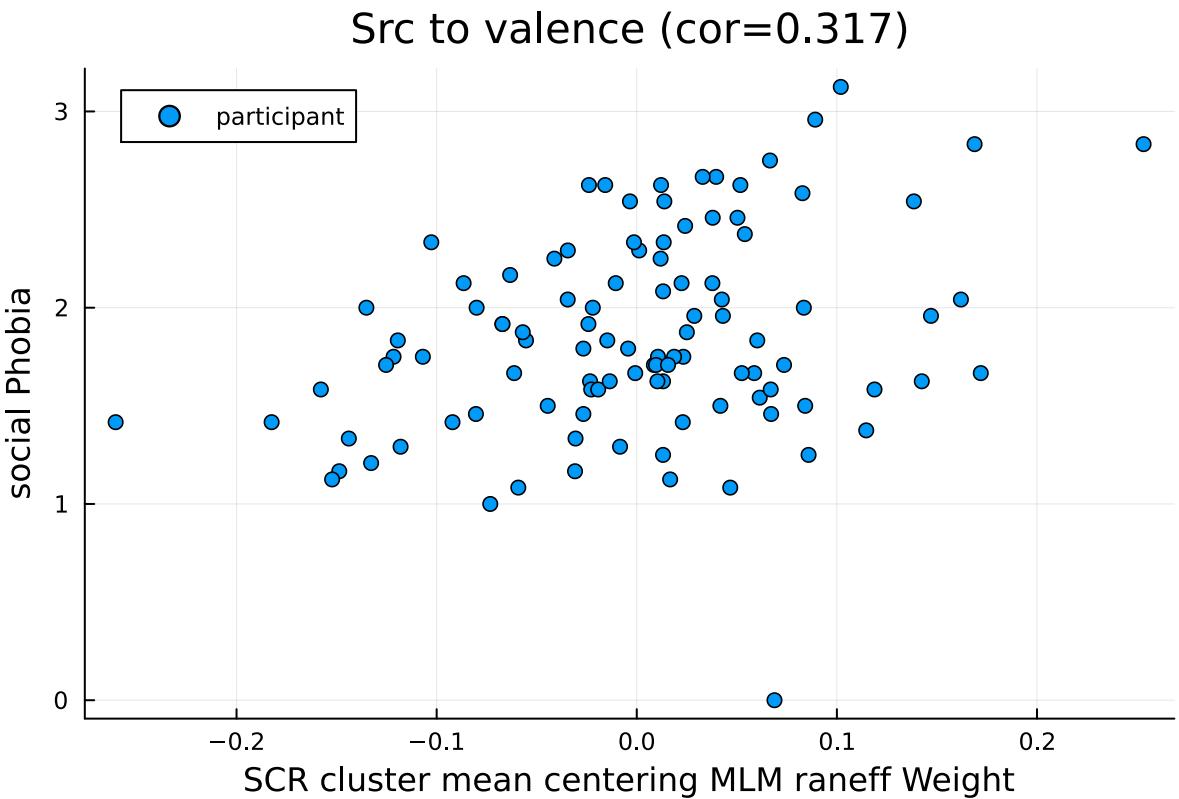
```
1 md"#### Scr to arousal, AIC: $(aic(mm12))"
```

```
1 mm12coef_i = cor(ranef(mm12)[1][1,:], df2_subjlv_hp.social_phobia);
```

```
1 mm12coef_w = cor(ranef(mm12)[1][2,:], df2_subjlv_hp.social_phobia);
```



```
1 scatter(ranef(mm12)[1][1,:], df2_subjlv_hp.social_phobia, label="participant",  
title="Src to valence (cor=$(get_trunc(mm12coef_i)))", xlabel="SCR cluster mean  
centering MLM raneff Intercept", ylabel="social Phobia")
```

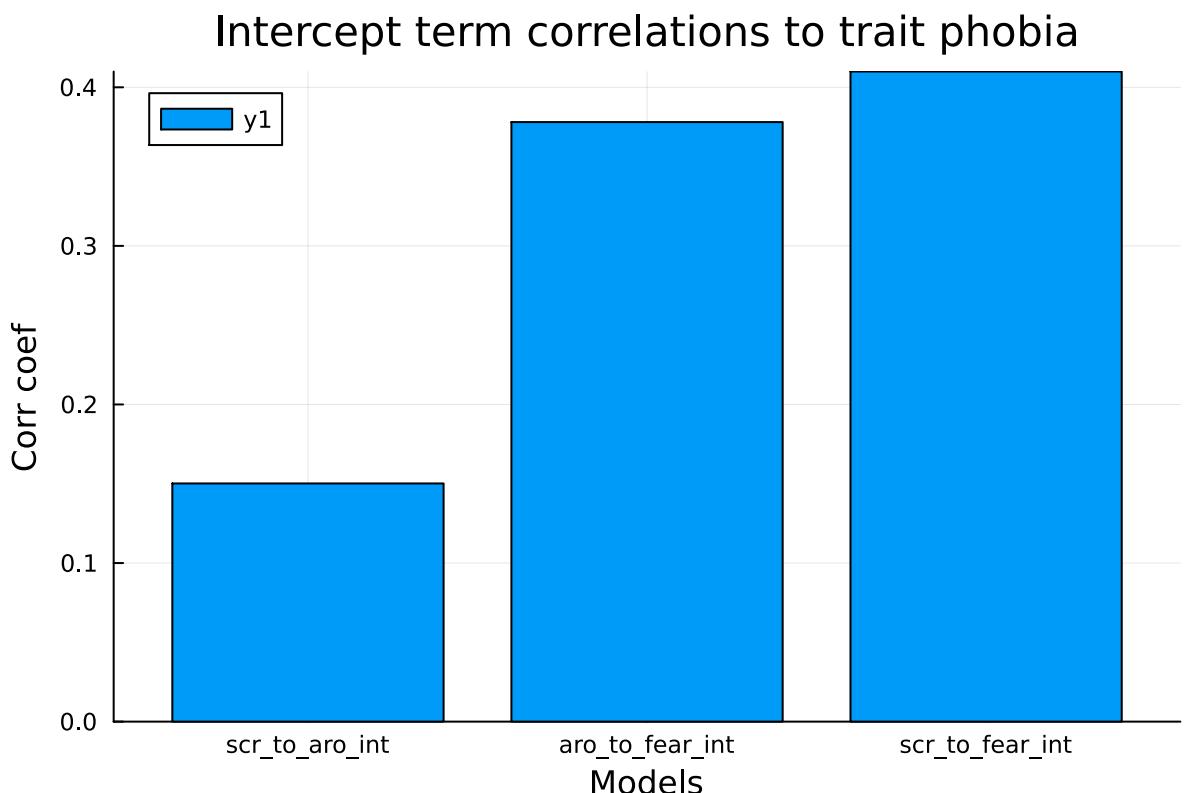


Model correlation comparison plots

```

1 md## Model correlation comparison plots"
1 plotdata3 = [mm3coef_i,mm4coef_i,mm6coef_i];
1 plotcols3 = ["scr_to_aro_int","aro_to_fear_int", "scr_to_fear_int"];

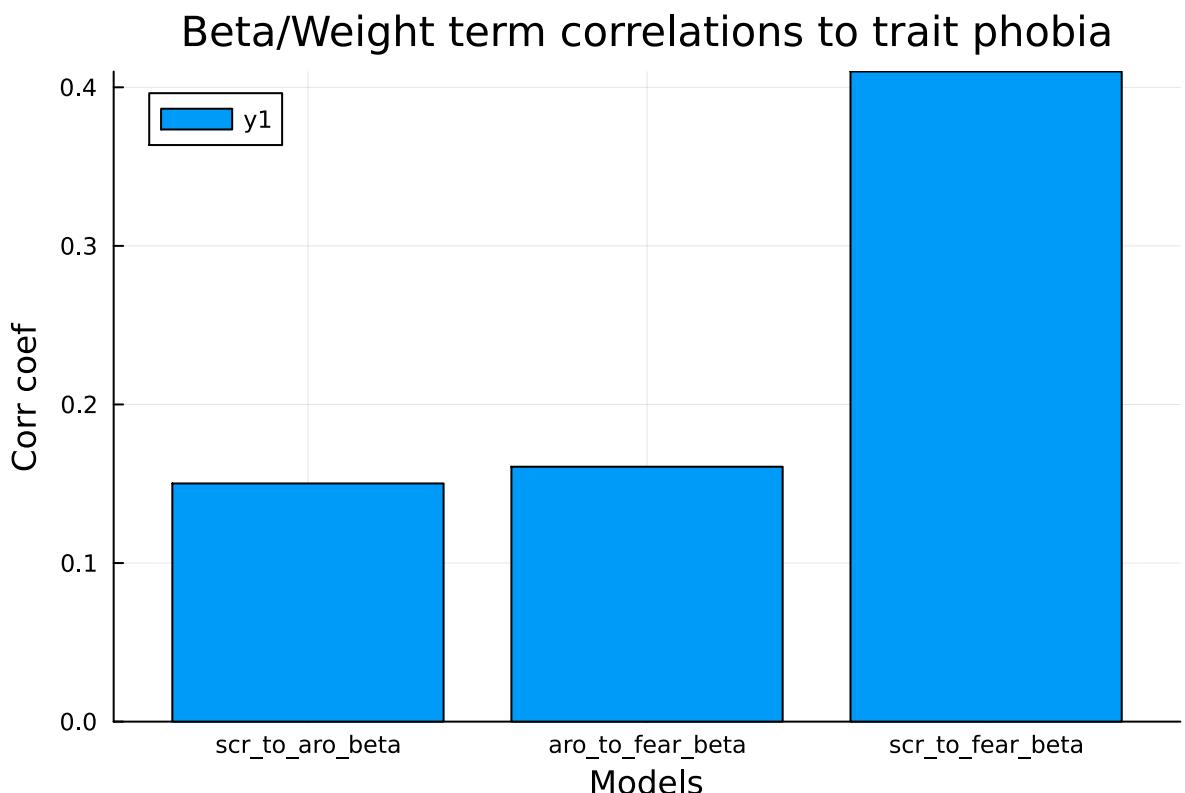
```



```
1 plot(bar(plotcols3, plotdata3), title="Intercept term correlations to trait phobia", xlabel="Models", ylabel="Corr coef")
```

```
1 plotdata4 = [mm3coef_w,mm4coef_w,mm6coef_w];
```

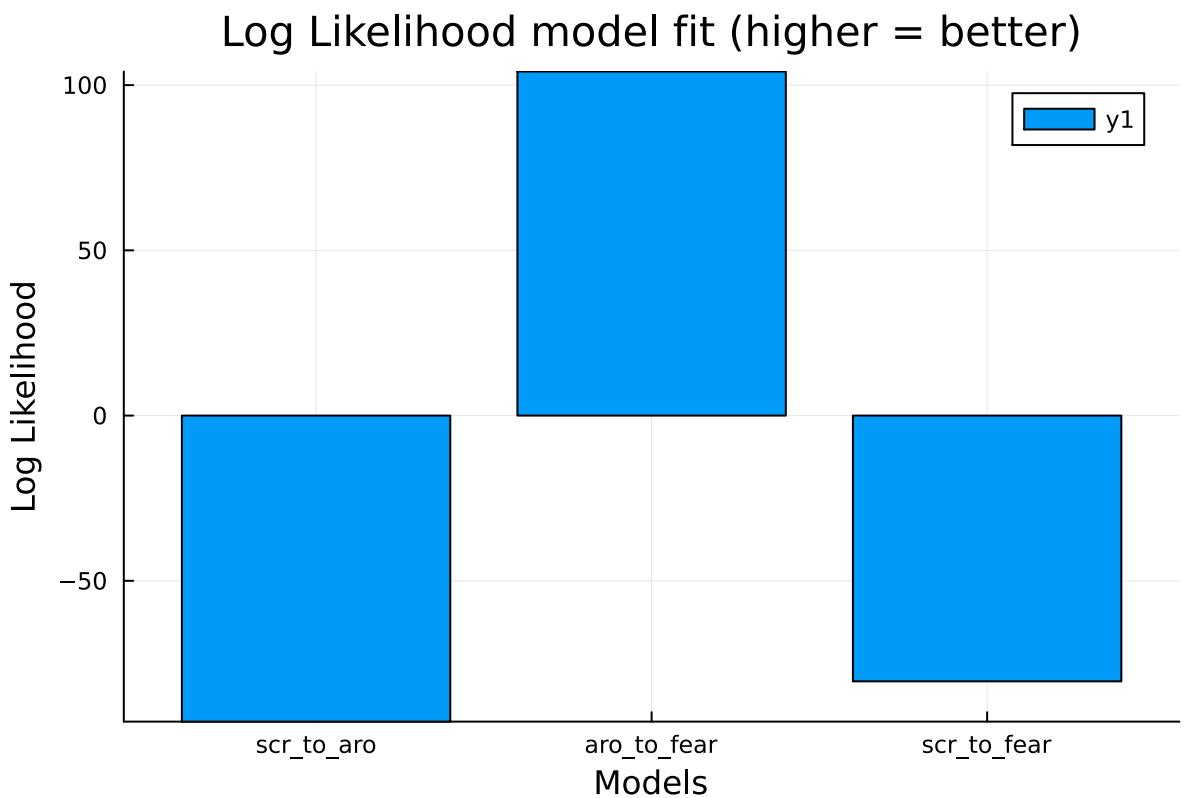
```
1 plotcols4 = ["scr_to_aro_beta","aro_to_fear_beta", "scr_to_fear_beta"];
```



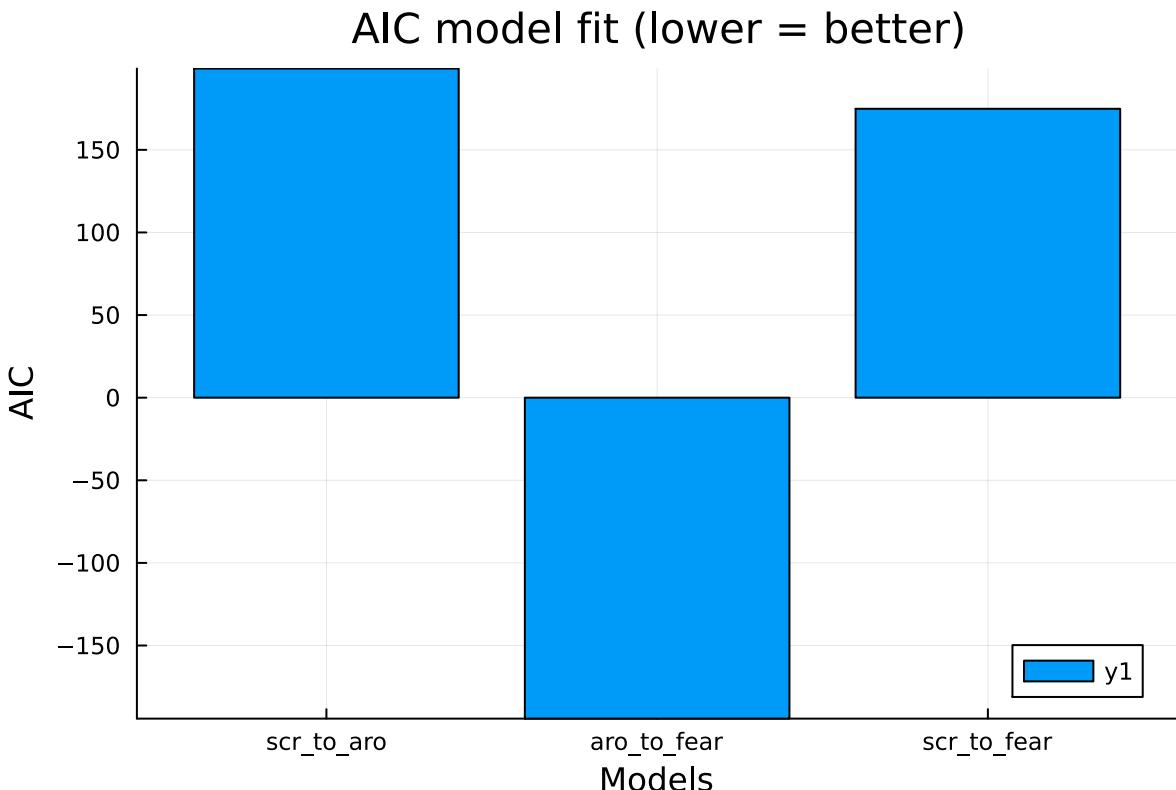
```
1 plot(bar(plotcols4, plotdata4), title="Beta/Weight term correlations to trait  
phobia", xlabel="Models", ylabel="Corr coef")
```

```
1 models = [mm3,mm4,mm6];
```

```
1 plotcols5 = ["scr_to_aro","aro_to_fear", "scr_to_fear"];
```



```
1 plot(bar(plotcols5, models .|> loglikelihood), title="Log Likelihood model fit  
(higher = better)", xlabel="Models", ylabel="Log Likelihood")
```



```
1 plot(bar(plotcols5, models .|> aic), title="AIC model fit (lower = better)",  
      xlabel="Models", ylabel="AIC")
```

Extra models

Scr change to Arousal

```
mm5 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.4658	0.0188	24.80	<1e-99	0.1642
scr_change_video_cmc	0.0107	0.2091	0.05	0.9591	0.6942
scr_change_video_mean	-0.4108	0.6165	-0.67	0.5052	
Residual	0.2407				

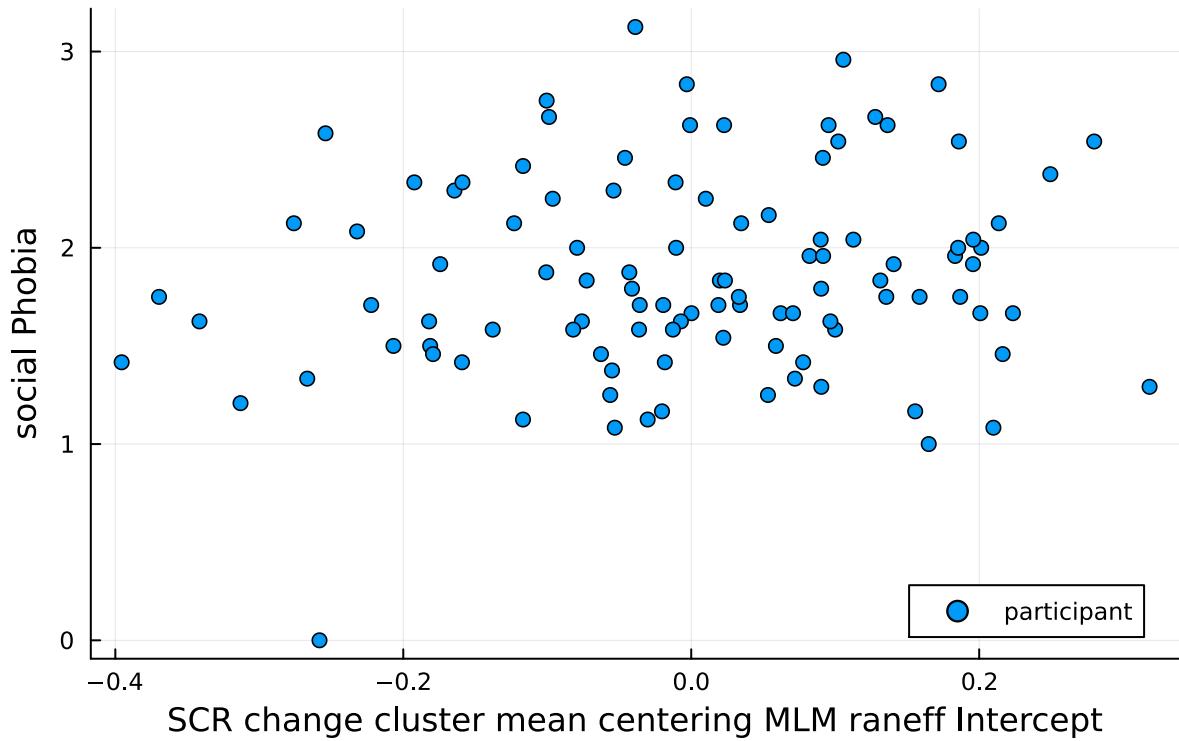
```
1 mm5 = fit(LinearMixedModel, @formula(resp_arousal ~ scr_change_video_cmc +  
      scr_change_video_mean + (scr_change_video_cmc|participant_num_str)), dfcleancol2)
```

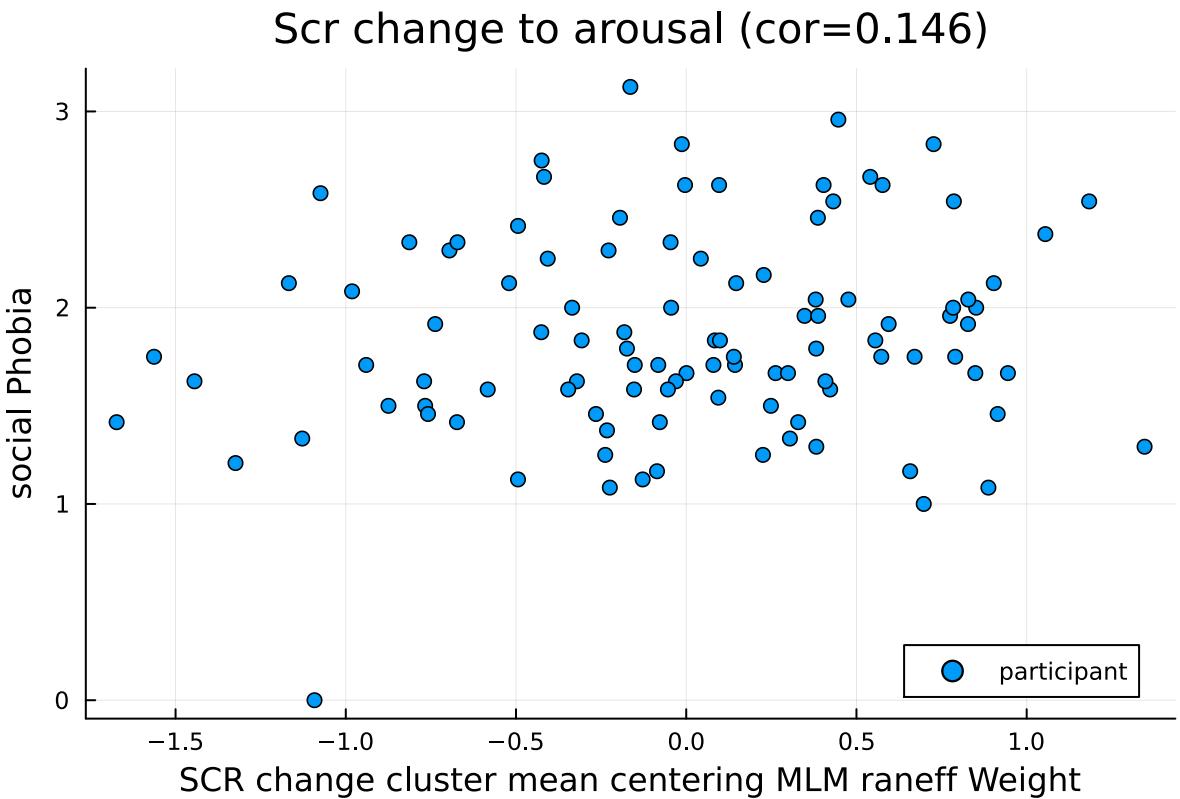
-92.10221772408102

```
1 loglikelihood(mm5)
```

198.20443544816203

Scr change to arousal (cor=0.146)





Scr change to Fear

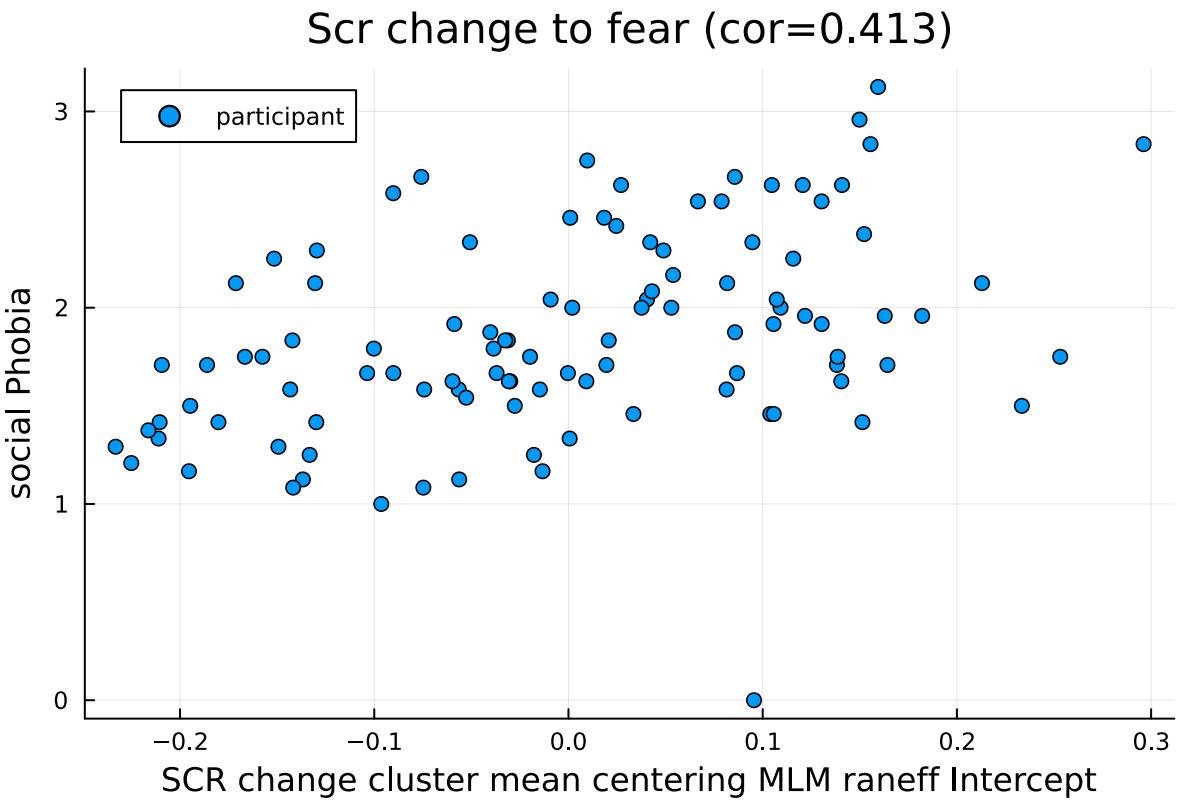
```
mm7 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.3123	0.0163	19.14	<1e-80	0.1370
scr_change_video_cmc	0.1890	0.2001	0.94	0.3449	0.3188
scr_change_video_mean	0.9139	0.5565	1.64	0.1005	
Residual	0.2416				

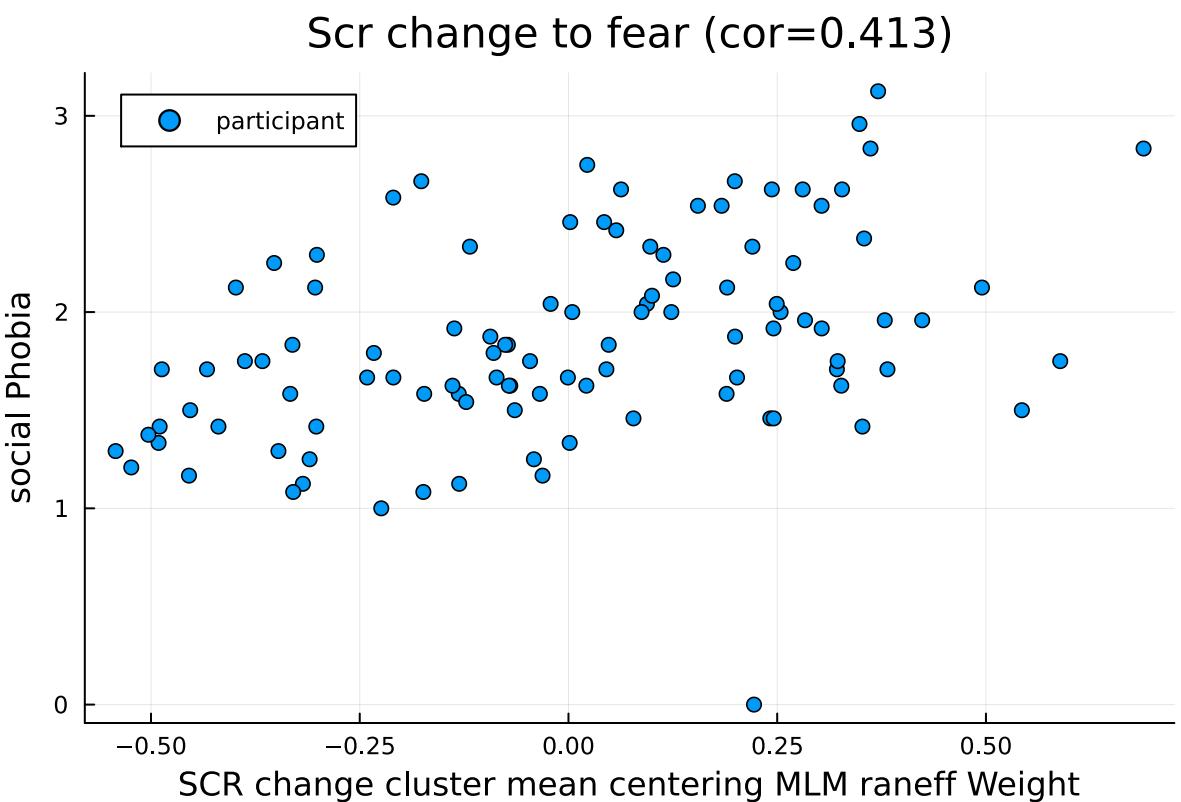
```
1 mm7 = fit(LinearMixedModel, @formula(resp_fear ~ scr_change_video_cmc +
scr_change_video_mean + (scr_change_video_cmc|participant_num_str)), dfcleancol2)
```

-80.01586125524496

```
1 mm7coef_i = cor(ranef(mm7)[1][1,:], df2_subjlvl_hp.social_phobia);
```



```
1 scatter(ranef(mm7)[1][1,:], df2_subjlvl_hp.social_phobia, label="participant",
  title="Scr change to fear (cor=$(get_trunc(mm7coef_i)))", xlabel="SCR change
  cluster mean centering MLM raneff Intercept", ylabel="social Phobia")
```



Scr to Arousal with social phobia modulating term

```
mm8 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.3623	0.0739	4.90	<1e-06	0.1624
video_scr_cmc	-1.1040	0.7756	-1.42	0.1546	0.6152
social_phobia	0.0515	0.0344	1.50	0.1343	
video_scr_mean	0.4230	0.8209	0.52	0.6064	
video_scr_cmc & social_phobia	0.6559	0.4054	1.62	0.1057	
Residual	0.2407				

```
1 mm8 = fit(LinearMixedModel, @formula(resp_arousal ~ video_scr_cmc * social_phobia + video_scr_mean + (video_scr_cmc|participant_num_str)), dfcleancol2)
```

-90.71485103035165

```
1 loglikelihood(mm8)
```

199.4297020607033

```
1 aic(mm8)
```

mm8coef_i = -4.59360597749135e-15

```
1 mm8coef_i = cor(ranef(mm8)[1][1,:], df2_subjlvl_hp.social_phobia)
```

mm8coef_w = -4.6278088290588755e-15

```
1 mm8coef_w = cor(ranef(mm8)[1][2,:], df2_subjlvl_hp.social_phobia)
```

```
mm9 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	-0.0550	0.0552	-1.00	0.3186	0.1197
resp_arousal_cmc	0.3342	0.2063	1.62	0.1053	0.4433
social_phobia	0.1127	0.0257	4.39	<1e-04	
resp_arousal_mean	0.3129	0.0662	4.73	<1e-05	
resp_arousal_cmc & social_phobia	0.0611	0.1068	0.57	0.5673	
Residual	0.1962				

```
1 mm9= fit(LinearMixedModel, @formula(resp_fear ~ resp_arousal_cmc * social_phobia + resp_arousal_mean + (resp_arousal_cmc|participant_num_str)), dfcleancol2)
```

```
113.8802912329055
```

```
1 loglikelihood(mm9)
```

```
mm10 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.0605	0.0596	1.01	0.3107	0.1232
video_scr_cmc	-0.6911	0.7471	-0.93	0.3550	0.2679
social_phobia	0.1284	0.0276	4.65	<1e-05	
video_scr_mean	0.0940	0.6750	0.14	0.8893	
video_scr_cmc & social_phobia	0.5063	0.3912	1.29	0.1956	
Residual	0.2414				

```
1 mm10= fit(LinearMixedModel, @formula(resp_fear ~ video_scr_cmc * social_phobia + video_scr_mean + (video_scr_cmc|participant_num_str)), dfcleancol2)
```

```
-70.3616941596519
```

```
1 loglikelihood(mm10)
```

```
mm11 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	-0.0472	0.0596	-0.79	0.4284	0.1196
video_scr_cmc	-0.0947	0.6436	-0.15	0.8830	0.3635
social_phobia	0.1112	0.0260	4.28	<1e-04	
resp_arousal_cmc	0.3441	0.2082	1.65	0.0984	0.4486
video_scr_mean	-0.1386	0.5636	-0.25	0.8058	
resp_arousal_mean	0.3101	0.0662	4.69	<1e-05	
video_scr_cmc & social_phobia	0.1087	0.3378	0.32	0.7476	
video_scr_cmc & resp_arousal_cmc	-1.2496	3.5119	-0.36	0.7220	2.1290
social_phobia & resp_arousal_cmc	0.0565	0.1078	0.52	0.5999	
video_scr_cmc & social_phobia & resp_arousal_cmc	1.0590	1.8771	0.56	0.5726	
Residual	0.1948				

```
1 mm11= fit(LinearMixedModel, @formula(resp_fear ~ video_scr_cmc * social_phobia * resp_arousal_cmc + video_scr_mean + resp_arousal_mean + (video_scr_cmc * resp_arousal_cmc|participant_num_str)), dfcleancol2)
```

```
Minimizing 1231    Time: 0:00:00 (81.23 µs/it)
objective: -232.48749836162784
```

```
Minimizing 1540    Time: 0:00:00 (77.27 µs/it)
```

```
Minimizing 1541    Time: 0:00:00 (77.87 µs/it)
objective: -232.48844971640148
```

mm13 =

	Est.	SE	z	p	σ_participant_num_str
(Intercept)	0.1039	0.0802	1.30	0.1950	0.1216
social_phobia	0.1029	0.0261	3.94	<1e-04	
video_scr_cmc	-0.1492	0.4961	-0.30	0.7637	0.2525
resp_arousal_cmc	0.2222	0.1273	1.75	0.0809	0.2102
resp_valence_cmc	-0.3115	0.1079	-2.89	0.0039	0.2216
video_scr_mean	0.1719	0.4834	0.36	0.7221	
resp_arousal_mean	0.1915	0.0566	3.38	0.0007	
resp_valence_mean	-0.1719	0.0898	-1.91	0.0555	
social_phobia & video_scr_cmc	0.0317	0.2632	0.12	0.9043	
social_phobia & resp_arousal_cmc	0.0698	0.0666	1.05	0.2946	
video_scr_cmc & resp_arousal_cmc	-0.0999	2.8027	-0.04	0.9716	1.3894
social_phobia & resp_valence_cmc	-0.0913	0.0573	-1.59	0.1109	
video_scr_cmc & resp_valence_cmc	3.1612	2.2510	1.40	0.1602	1.5476
resp_arousal_cmc & resp_valence_cmc	-0.7601	0.3964	-1.92	0.0552	0.5847
social_phobia & video_scr_cmc & resp_arousal_cmc	0.3158	1.5294	0.21	0.8364	
social_phobia & video_scr_cmc & resp_valence_cmc	-1.4947	1.2412	-1.20	0.2285	
social_phobia & resp_arousal_cmc & resp_valence_cmc	0.2527	0.2079	1.22	0.2242	
video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	12.9223	9.9970	1.29	0.1961	2.1339
social_phobia & video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	-6.0031	5.4359	-1.10	0.2695	
Residual	0.1445				

```
1 mm13= fit(LinearMixedModel, @formula(resp_fear ~ social_phobia * video_scr_cmc *  
+ resp_arousal_cmc * resp_valence_cmc + video_scr_mean + resp_arousal_mean +  
+ resp_valence_mean + (video_scr_cmc * resp_arousal_cmc *  
+ resp_valence_cmc|participant_num_str)), dfcleancol2)
```



Minimizing 575 Time: 0:00:00 (0.18 ms/it)



```
objective: -864.2518897234419

Minimizing 1286    Time: 0:00:00 ( 0.16 ms/it)
objective: -865.9478724742362

Minimizing 1990    Time: 0:00:00 ( 0.16 ms/it)
objective: -866.6883601138629

Minimizing 2687    Time: 0:00:00 ( 0.16 ms/it)
objective: -867.0616183762538

Minimizing 3363    Time: 0:00:00 ( 0.16 ms/it)
objective: -867.1661829221365

Minimizing 4073    Time: 0:00:00 ( 0.16 ms/it)
objective: -867.2100263542038

Minimizing 4771    Time: 0:00:00 ( 0.16 ms/it)
objective: -867.2567851229655
```

116.24422485820074

```
1 loglikelihood(mm11)
```

433.6525574061808

```
1 loglikelihood(mm13)
```

-190.48844971640148

```
1 aic(mm11)
```

-755.3051148123616

```
1 aic(mm13)
```

mm14 =

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.1195	0.0828	1.44	0.1493	0.1213
social_phobia	0.1015	0.0260	3.91	<1e-04	
video_scr_cmc	-0.1524	0.4999	-0.30	0.7604	0.1911
video_hp_cmc	-0.0691	0.1180	-0.59	0.5583	0.0771
resp_arousal_cmc	0.2454	0.1279	1.92	0.0551	0.2142
resp_valence_cmc	-0.3022	0.1116	-2.71	0.0068	0.2295
video_scr_mean	0.2773	0.4756	0.58	0.5598	
video_hp_mean	-0.0533	0.0363	-1.47	0.1418	
resp_arousal_mean	0.2327	0.0538	4.32	<1e-04	
resp_valence_mean	-0.1670	0.0878	-1.90	0.0570	
social_phobia & video_scr_cmc	0.0361	0.2645	0.14	0.8915	
social_phobia & video_hp_cmc	0.0235	0.0616	0.38	0.7029	
video_scr_cmc & video_hp_cmc	2.5321	2.9122	0.87	0.3846	1.1369
social_phobia & resp_arousal_cmc	0.0614	0.0668	0.92	0.3581	
video_scr_cmc & resp_arousal_cmc	0.4776	2.8608	0.17	0.8674	1.2606
video_hp_cmc & resp_arousal_cmc	0.6455	0.6513	0.99	0.3216	0.4485
social_phobia & resp_valence_cmc	-0.0989	0.0593	-1.67	0.0954	
video_scr_cmc & resp_valence_cmc	3.2593	2.3563	1.38	0.1666	1.4579
video_hp_cmc & resp_valence_cmc	0.8944	0.4774	1.87	0.0610	0.2998
resp_arousal_cmc & resp_valence_cmc	-0.9318	0.3833	-2.43	0.0150	0.5144
social_phobia & video_scr_cmc & video_hp_cmc	-1.0762	1.5689	-0.69	0.4927	
social_phobia & video_scr_cmc & resp_arousal_cmc	-0.1044	1.5700	-0.07	0.9470	
social_phobia & video_hp_cmc & resp_arousal_cmc	-0.4342	0.3185	-1.36	0.1728	
video_scr_cmc & video_hp_cmc & resp_arousal_cmc	10.7027	14.9570	0.72	0.4743	3.5659
social_phobia & video_scr_cmc & resp_valence_cmc	-1.4930	1.3102	-1.14	0.2545	

social_phobia & video_hp_cmc & resp_valence_cmc	-0.4326	0.2347	-1.84	0.0653	
video_scr_cmc & video_hp_cmc & resp_valence_cmc	13.6269	12.0520	1.13	0.2582	2.6077
social_phobia & resp_arousal_cmc & resp_valence_cmc	0.3279	0.2021	1.62	0.1046	
video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	11.2366	11.3060	0.99	0.3203	2.1820
video_hp_cmc & resp_arousal_cmc & resp_valence_cmc	-0.3942	2.3712	-0.17	0.8680	1.3916
social_phobia & video_scr_cmc & video_hp_cmc & resp_arousal_cmc	-4.6430	7.6596	-0.61	0.5444	
social_phobia & video_scr_cmc & video_hp_cmc & resp_valence_cmc	-6.4850	6.1237	-1.06	0.2896	
social_phobia & video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	-4.8392	6.2947	-0.77	0.4420	
social_phobia & video_hp_cmc & resp_arousal_cmc & resp_valence_cmc	0.4002	1.1561	0.35	0.7292	
video_scr_cmc & video_hp_cmc & resp_arousal_cmc & resp_valence_cmc	55.8137	57.1621	0.98	0.3289	1.8273
social_phobia & video_scr_cmc & video_hp_cmc & resp_arousal_cmc & resp_valence_cmc	-28.3001	26.8477	-1.05	0.2918	
Residual	0.1423				

```
1 mm14 = fit(LinearMixedModel, @formula(resp_fear ~ social_phobia * video_scr_cmc
* video_hp_cmc * resp_arousal_cmc * resp_valence_cmc + video_scr_mean +
video_hp_mean + resp_arousal_mean + resp_valence_mean + (video_scr_cmc *
video_hp_cmc * resp_arousal_cmc * resp_valence_cmc|participant_num_str)),
dfcleancol2)
```

Minimizing 246 Time: 0:00:00 (0.41 ms/it)
 objective: -783.0527897560318

```
Minimizing 491      Time: 0:00:00 ( 0.63 ms/it)
objective: -871.6028354416678
```

```
Minimizing 608      Time: 0:00:00 ( 0.68 ms/it)
objective: -874.8441885775587
```

```
Minimizing 724      Time: 0:00:00 ( 0.71 ms/it)
objective: -876.7009190397433
```

```
Minimizing 836      Time: 0:00:00 ( 0.75 ms/it)
objective: -878.8865396211761
```

```
Minimizing 945      Time: 0:00:00 ( 0.77 ms/it)
objective: -880.0192307162106
```

```
Minimizing 1050     Time: 0:00:00 ( 0.79 ms/it)
objective: -881.7608116173526
```

```
mm15 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.1667	0.0874	1.91	0.0564	0.1180
social_phobia	0.1000	0.0255	3.92	<1e-04	
video_scr_cmc	-0.4310	0.4869	-0.89	0.3761	0.1365
video_hp_cmc	-0.1360	0.1011	-1.35	0.1784	0.0337
resp_arousal_cmc	0.1384	0.1148	1.21	0.2280	0.1779
resp_valence_cmc	-0.3231	0.1049	-3.08	0.0021	0.2092
video_scr_mean	0.3121	0.5111	0.61	0.5414	
video_hp_mean	-0.0785	0.0397	-1.97	0.0483	
resp_arousal_mean	0.2465	0.0585	4.22	<1e-04	
resp_valence_mean	-0.2306	0.0958	-2.41	0.0160	
social_phobia & video_scr_cmc	0.1876	0.2545	0.74	0.4609	
social_phobia & video_hp_cmc	0.0684	0.0517	1.32	0.1855	
social_phobia & resp_arousal_cmc	0.1004	0.0601	1.67	0.0951	
social_phobia & resp_valence_cmc	-0.0846	0.0559	-1.52	0.1297	
Residual	0.1521				

```
1 mm15 = fit(LinearMixedModel, @formula(resp_fear ~ social_phobia * video_scr_cmc + social_phobia * video_hp_cmc + social_phobia * resp_arousal_cmc + social_phobia * resp_valence_cmc + video_scr_mean + video_hp_mean + resp_arousal_mean + resp_valence_mean + (video_scr_cmc + video_hp_cmc + resp_arousal_cmc + resp_valence_cmc | participant_num_str)), dfcleancol2)
```

```
> Minimizing 895    Time: 0:00:00 ( 0.11 ms/it)
  objective: -815.8719822315832

Minimizing 1011    Time: 0:00:00 ( 0.11 ms/it)

Minimizing 1012    Time: 0:00:00 ( 0.11 ms/it)
  objective: -815.8720929432236
```

mm16 =

	Est.	SE	z	p	σ_participant_num_str
(Intercept)	0.1321	0.0825	1.60	0.1096	0.1251
social_phobia	0.0937	0.0245	3.82	0.0001	
video_scr_cmc	-0.2476	0.4772	-0.52	0.6039	0.2265
video_hp_cmc	-0.1097	0.0972	-1.13	0.2589	0.0648
resp_arousal_cmc	0.0786	0.1120	0.70	0.4827	0.2156
resp_valence_cmc	-0.3464	0.1070	-3.24	0.0012	0.2249
video_scr_mean	0.1294	0.4762	0.27	0.7858	
video_hp_mean	-0.0450	0.0364	-1.24	0.2157	
resp_arousal_mean	0.2347	0.0546	4.30	<1e-04	
resp_valence_mean	-0.1600	0.0888	-1.80	0.0715	
social_phobia & video_scr_cmc	0.0693	0.2489	0.28	0.7808	
social_phobia & video_hp_cmc	0.0570	0.0496	1.15	0.2503	
social_phobia & resp_arousal_cmc	0.1244	0.0584	2.13	0.0332	
social_phobia & resp_valence_cmc	-0.0787	0.0569	-1.38	0.1664	
resp_arousal_cmc & resp_valence_cmc					0.6384
video_hp_cmc & resp_arousal_cmc & resp_valence_cmc					1.3278
video_scr_cmc & video_hp_cmc & resp_valence_cmc					2.8790
video_scr_cmc & video_hp_cmc					0.8911
video_scr_cmc & video_hp_cmc & resp_arousal_cmc					3.1346
video_scr_cmc & resp_valence_cmc					1.6609
video_scr_cmc & resp_arousal_cmc					1.2975
video_hp_cmc & resp_arousal_cmc					0.3361
video_scr_cmc & resp_arousal_cmc & resp_valence_cmc					2.2885
video_scr_cmc & video_hp_cmc & resp_arousal_cmc & resp_valence_cmc					2.7962
video_hp_cmc &					0.2961

```
resp_valence_cmc  
Residual          0.1432
```

```
1 mm16 = fit(LinearMixedModel, @formula(resp_fear ~ social_phobia * video_scr_cmc  
+ social_phobia * video_hp_cmc + social_phobia * resp_arousal_cmc +  
social_phobia * resp_valence_cmc + video_scr_mean + video_hp_mean +  
resp_arousal_mean + resp_valence_mean + (video_scr_cmc * video_hp_cmc *  
resp_arousal_cmc * resp_valence_cmc|participant_num_str)), dfcleancol2)
```

```
Minimizing 436    Time: 0:00:00 ( 0.45 ms/it)  
objective: -848.7715997277515

Minimizing 657    Time: 0:00:00 ( 0.57 ms/it)  
objective: -852.0340395259268

Minimizing 782    Time: 0:00:00 ( 0.62 ms/it)  
objective: -853.4885343084786

Minimizing 900    Time: 0:00:00 ( 0.66 ms/it)  
objective: -855.300478724758

Minimizing 1017   Time: 0:00:00 ( 0.68 ms/it)  
objective: -857.0496319636221

Minimizing 1131   Time: 0:00:00 ( 0.70 ms/it)  
objective: -859.4252079529268

Minimizing 1244   Time: 0:00:00 ( 0.72 ms/it)  
objective: -861.6464766088978
```

Plot Test

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_a
1	"social_low_1.m4v"	107	0.046	2.003	0.052
2	"social_high_6.mov"	107	0.061	2.103	0.059
3	"social_low_6.m4v"	107	0.042	2.336	0.046
4	"social_high_3.mov"	107	0.034	2.353	0.048
5	"social_low_3.m4v"	107	0.052	1.902	0.055
6	"social_low_2.mov"	107	0.039	3.721	0.059
7	"social_high_2.mov"	107	0.046	2.136	0.063
8	"social_high_1.mov"	107	0.051	1.201	0.038
9	"social_low_4.m4v"	107	0.068	1.318	0.059
10	"social_low_5.m4v"	107	0.053	2.203	0.052
: more					
1224	"social_low_2.mov"	232	0.216	4.872	0.115

```
1 dfcleancol2
```

► [0.808, 0.757, 0.507, 0.608, 0.404, 0.272, 0.98, 0.804, 0.53, 0.694, 0.741, 0.751, 0.87]

```
1 dfcleancol2.resp_arousal
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.4623	0.0294	15.71	<1e-54	0.1644
video_scr_cmc	0.0992	0.2068	0.48	0.6314	0.6593
video_scr_mean	0.2780	0.8139	0.34	0.7327	
Residual	0.2408				

```
1 mm3
```

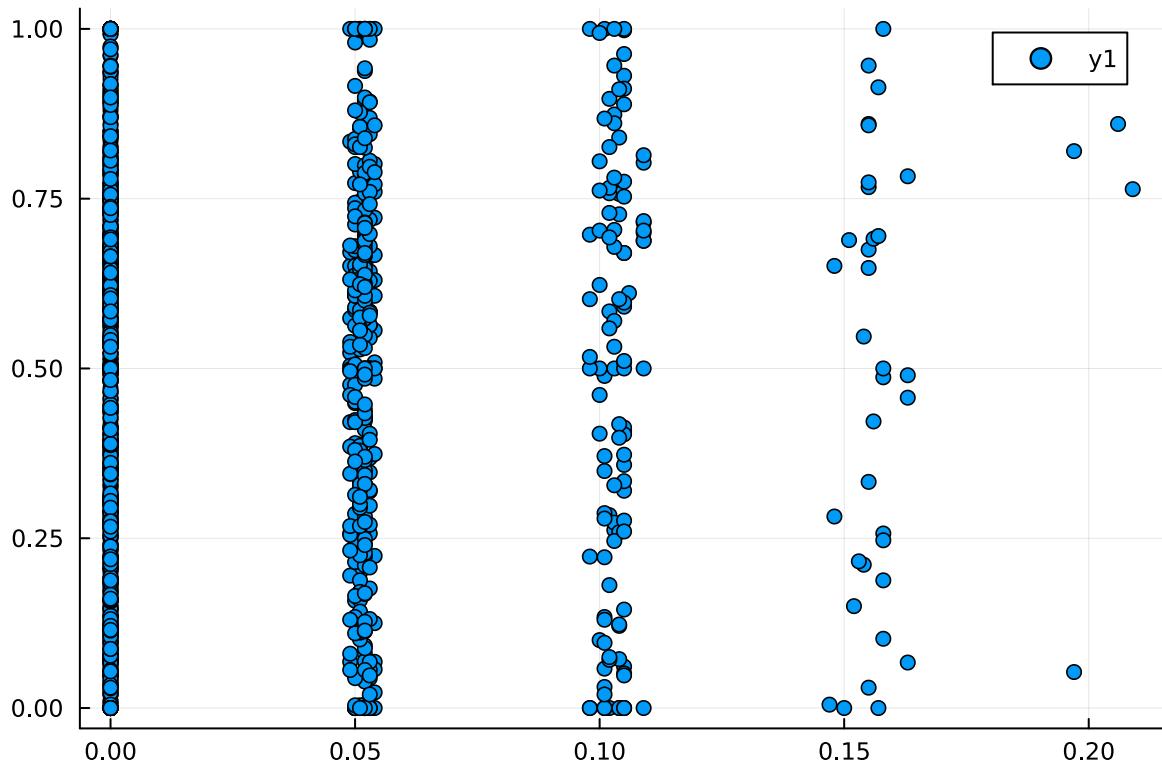
```
randeff =
2×105 Matrix{Float64}:
0.166288 0.0198479 0.148815 0.1944 ... 0.0867119 0.0952605 -0.121021
0.666734 0.0795807 0.596677 0.779452 ... 0.347673 0.381949 -0.485237
```

```
1 randeff = only(ranef(mm3))
```

```
► [0.166288, 0.0198479, 0.148815, 0.1944, 0.0375656, 0.134313, 0.00420013, -0.0413537, -  
1 randeff[1,:]
```

```
2×1 Matrix{Float64}:  
0.15197229090888623  
0.6093363217404658
```

```
1 std(only(ranef(mm3)), dims=2)
```



```
1 scatter(dfcleancol2.video_scr, dfcleancol2.resp_arousal)
```

```
0.209
```

```
1 maximum(dfcleancol2.video_scr)
```

```
0.0
```

```
1 minimum(dfcleancol2.video_scr)
```

```
xax =  
10-element LinRange{Float64, Int64}:  
0.0, 0.0232222, 0.0464444, 0.0696667, 0.0928889, 0.116111, 0.139333, 0.162556, 0.185778, 0.209  
1 xax = LinRange(minimum(dfcleancol2.video_scr),maximum(dfcleancol2.video_scr),10)
```

```
inter = 0.462344466012434
```

```
1 inter = coef(mm3)[1]
```

cmcbeta = 0.0992136005687378

```
1 cmcbeta = coef(mm3)[2]
```

meanbeta = 0.2779529251952862

```
1 meanbeta = coef(mm3)[3]
```

yax =

```
10-element LinRange{Float64, Int64}:
0.462344, 0.471103, 0.479862, 0.48862, 0.497379, ..., 0.514896, 0.523655, 0.532414, 0.541172
```

```
1 yax = inter .+ ((cmcbeta .* xax) + (meanbeta .* xax))
```

```
[1.0, 1.666667, 1.75, 2.125, 1.708333, 1.833333, 1.833333, 1.25, 1.916667, 1.708333, 1.333333]
```

1 df2_subjlvl_hp.social_phobia

0.5112248622557003

```
1 std(df2_subjlvl_hp.social_phobia)
```

```
std_up = 2.371542322573161
```

```
1 std_up = mean(df2_subjlv_hp.social_phobia) + std(df2_subjlv_hp.social_phobia)
```

bm_std_up =

```
► BitVector: [false, false, false, false, false, false, false, false, false, false, fals
```

```
1 bm_std_up = df2_subjlvl_hp.social_phobia .> std_up
```

0.04812321055798916

```
1 mean(randeff[1,bm_std_up])
```

std_dwn = 1.34909259806176

```
1 std_dwn = mean(df2_subjlvl_hp.social_phobia) - std(df2_subjlvl_hp.social_phobia)
```

bm_std_dwn =

► BitVector: [true, false, false, false, false, false, false, false, true, false, false, true, :

```
1 bm_std_dwn = df2_subjlvl_hp.social_phobia .< std_dwn
```

-0.006348771944764269

```
1 mean(randeff[1,bm_std_dwn])
```

yax_up =

10-element LinRange{Float64, Int64}:

0.0481232,0.052604,0.0570847,0.0615655,...,0.0750077,0.0794885,0.0839692,0.08845

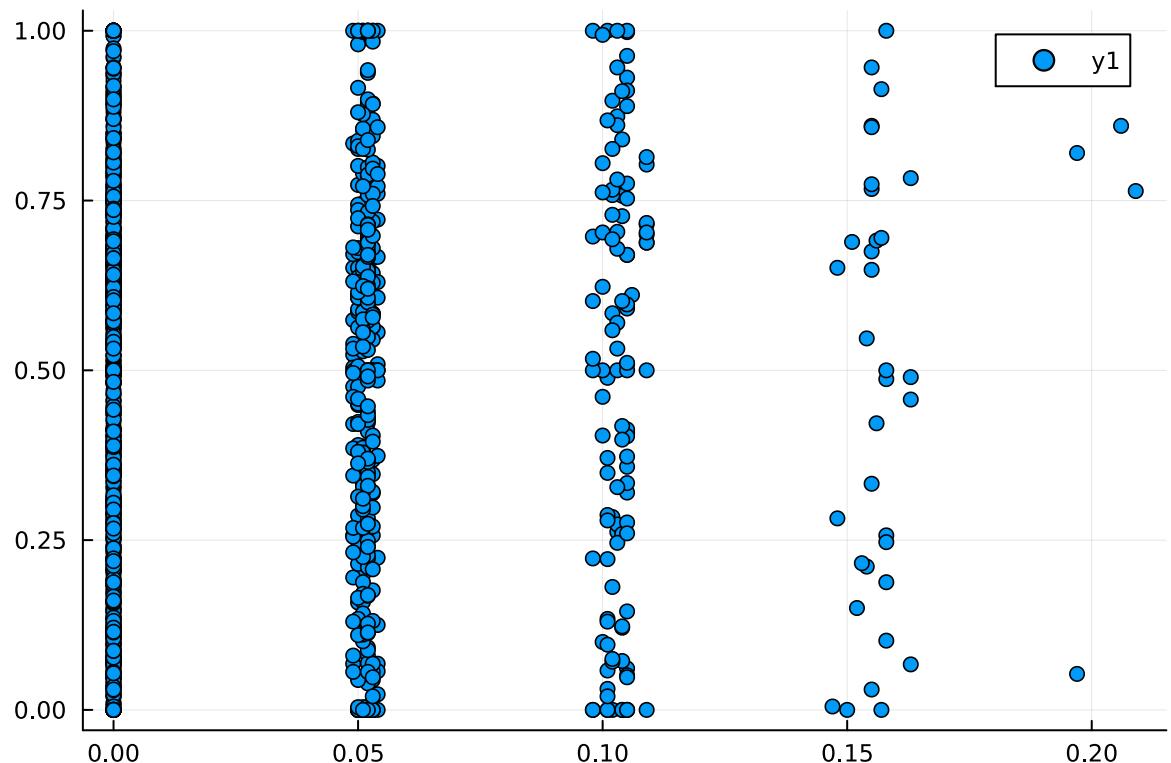
```
1 yax_up = mean(randeff[1,bm_std_up]) .+ (mean(randeff[2,bm_std_up]).* xax)
```

```
yax_dwn =  
► [0.00634877, 0.00693991, 0.00753104, 0.00812217, 0.00871331, 0.00930444, 0.00989558, 0  
1 yax_dwn = abs.(mean(randeff[1,bm_std_dwn]) .+ (mean(randeff[2,bm_std_dwn]).*  
xax))
```

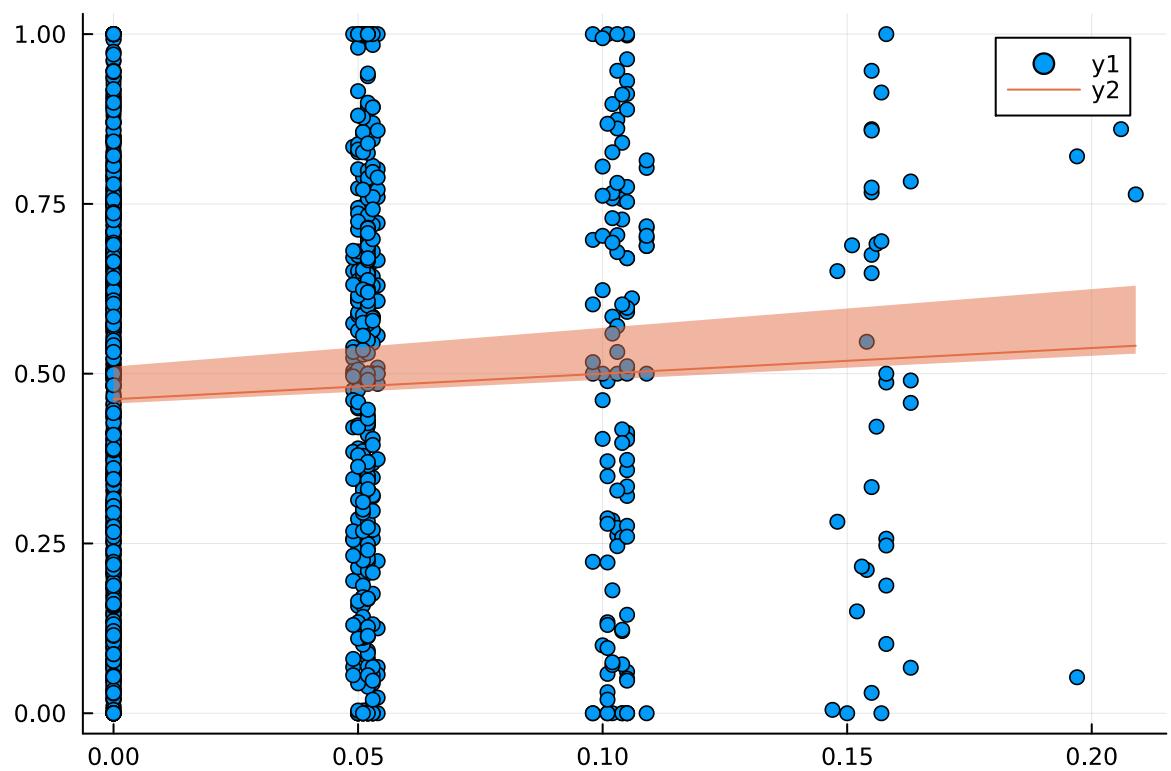
df =

	xax	yax	yax_up	yax_dwn
1	0.0	0.462344	0.0481232	0.00634877
2	0.0232222	0.471103	0.052604	0.00693991
3	0.0464444	0.479862	0.0570847	0.00753104
4	0.0696667	0.48862	0.0615655	0.00812217
5	0.0928889	0.497379	0.0660462	0.00871331
6	0.116111	0.506138	0.070527	0.00930444
7	0.139333	0.514896	0.0750077	0.00989558
8	0.162556	0.523655	0.0794885	0.0104867
9	0.185778	0.532414	0.0839692	0.0110778
10	0.209	0.541172	0.08845	0.011669

```
1 df = DataFrame(xax = xax, yax = yax, yax_up = yax_up, yax_dwn = yax_dwn)
```



```
1 scatter(dfcleancol2.video_scr, dfcleancol2.resp_arousal)
```



```
1 @df df plot!(:xax, :yax, ribbon=(:yax_dwn,:yax_up))
```

	Est.	SE	z	p	σ_participant_num_str
(Intercept)	0.4623	0.0294	15.71	<1e-54	0.1644
video_scr_cmc	0.0992	0.2068	0.48	0.6314	0.6593
video_scr_mean	0.2780	0.8139	0.34	0.7327	
Residual	0.2408				

1 mm3

▶ [2×105 Matrix{Float64}]:
 0.166288 0.0198479 0.148815 0.1944 ... 0.0867119 0.0952605 -0.121021]

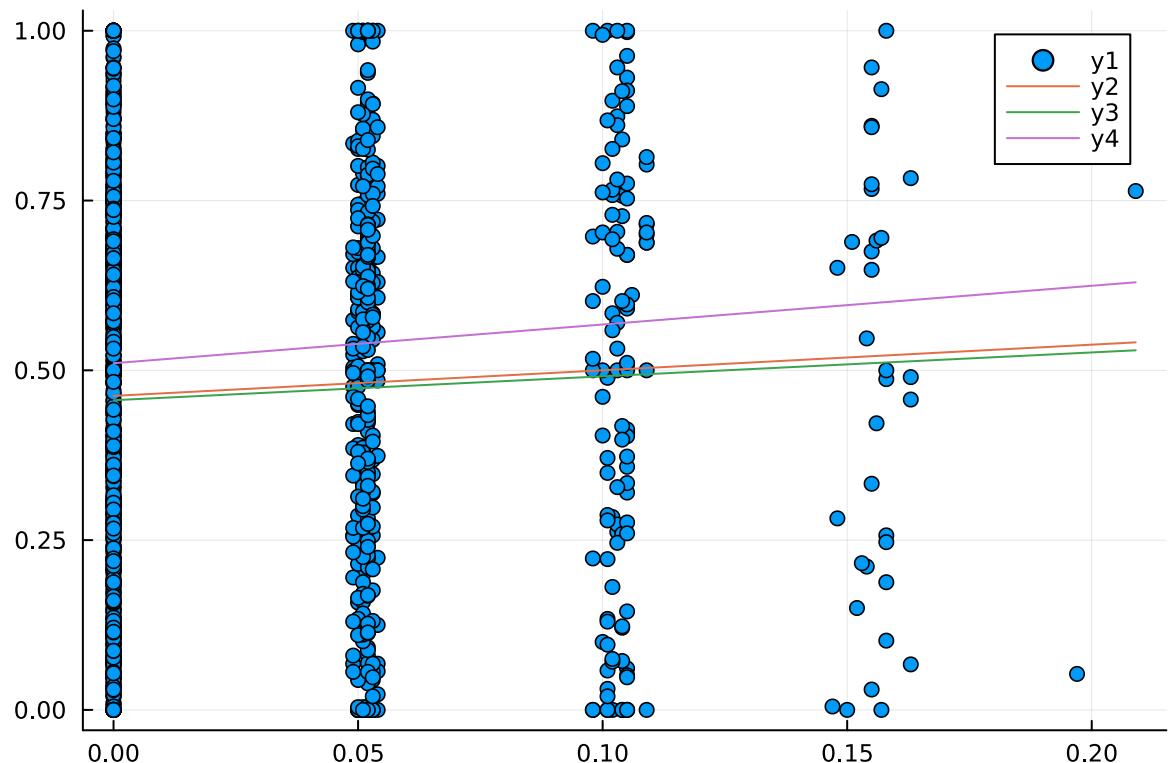
1 ranef(mm3)

1 Enter cell code...

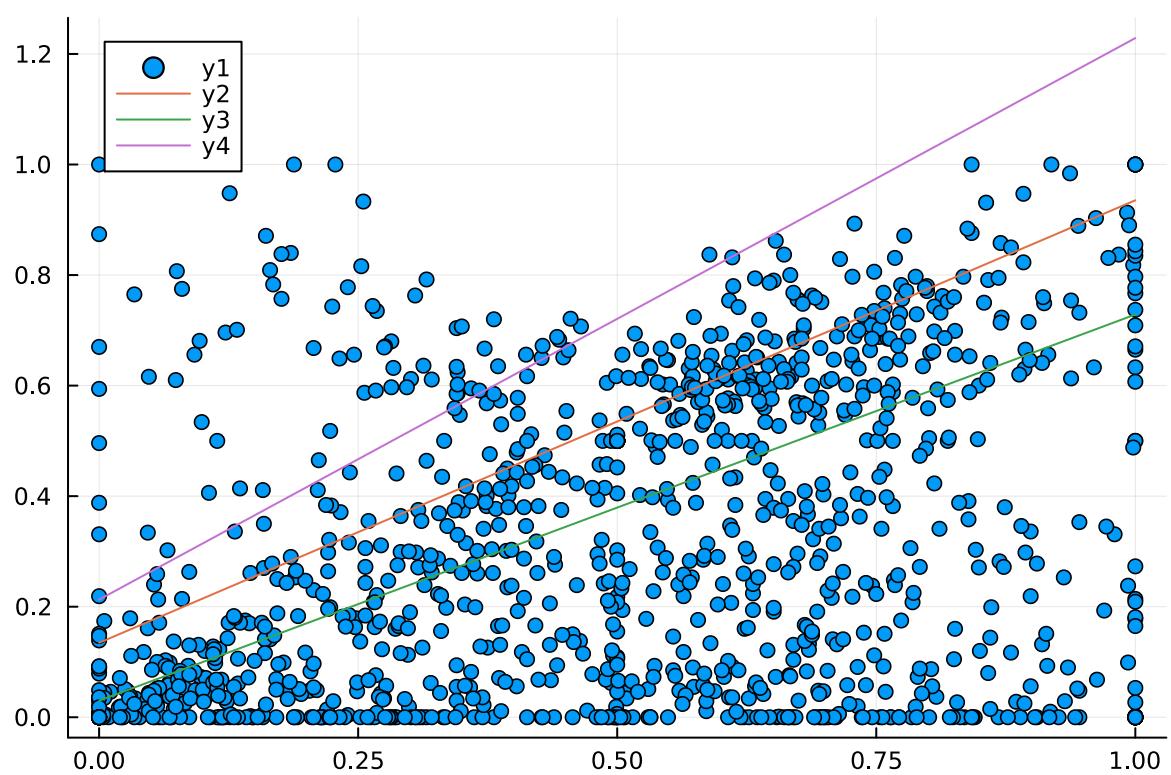
Plot Generic

plot_mm (generic function with 1 method)

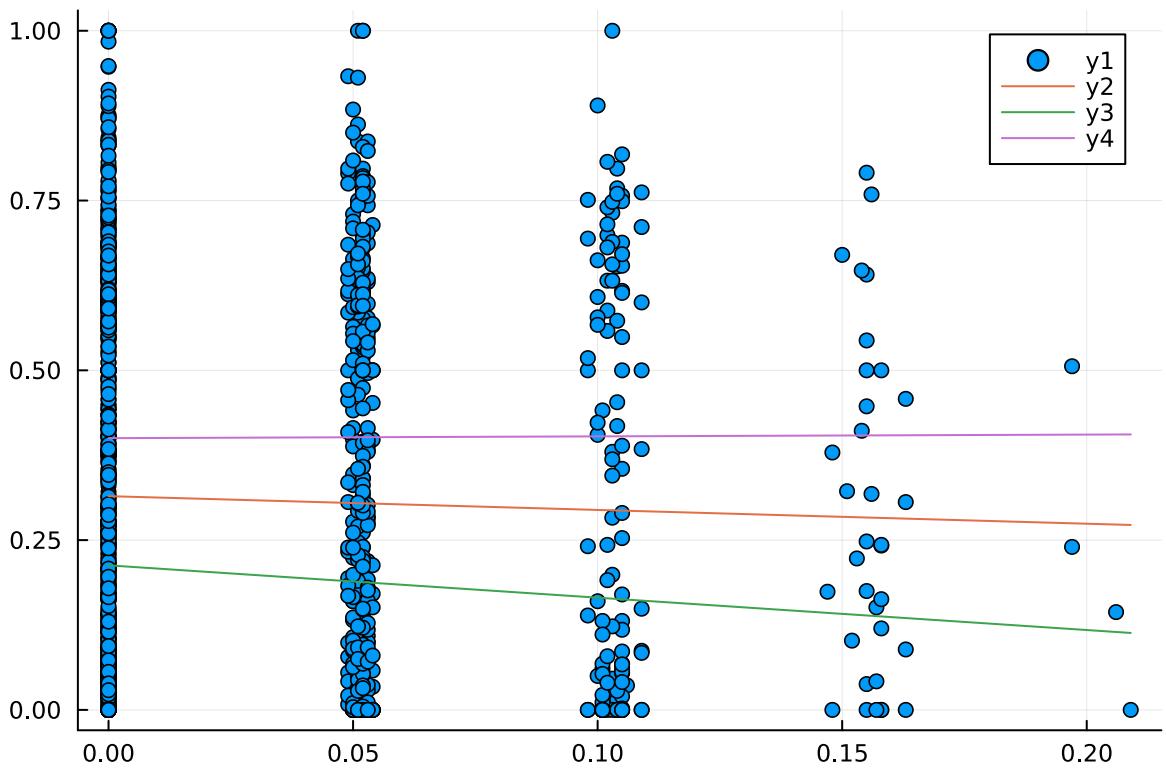
```
1 function plot_mm(mm, varx, vary)
2     randeff = only(ranef(mm))
3     xax = LinRange(minimum(varx),maximum(varx),10)
4     inter = coef(mm)[1]
5     cmcbeta = coef(mm)[2]
6     meanbeta = coef(mm)[3]
7     yax = inter .+ ((cmcbeta .* xax) + (meanbeta .* xax))
8     std_up = mean(df2_subjlvl_hp.social_phobia) +
9     std(df2_subjlvl_hp.social_phobia)
10    bm_std_up = df2_subjlvl_hp.social_phobia .> std_up
11    std_dwn = mean(df2_subjlvl_hp.social_phobia) -
12    std(df2_subjlvl_hp.social_phobia)
13    bm_std_dwn = df2_subjlvl_hp.social_phobia .< std_dwn
14    yax_up = yax + (mean(randeff[1,bm_std_up]) .+ (mean(randeff[2,bm_std_up]).* xax))
15    yax_dwn = yax + (mean(randeff[1,bm_std_dwn]) .+
16    (mean(randeff[2,bm_std_dwn]).* xax))
17    df = DataFrame(xax = xax, yax = yax, yax_up = yax_up, yax_dwn = yax_dwn)
18    scatter(varx, vary)
19    @df df plot!(:xax, :yax)
20    @df df plot!(:xax, :yax_dwn)
21    @df df plot!(:xax, :yax_up)
end
```



```
1 plot_mm(mm3, dfcleancol2.video_scr, dfcleancol2.resp_arousal)
```



```
1 plot_mm(mm4, dfcleancol2.resp_arousal, dfcleancol2.resp_fear)
```



```
1 plot_mm(mm6, dfcleancol2.video_scr, dfcleancol2.resp_fear)
```

Mixed Models - With z-scores

UndefVarError: df2cleancol not defined

1. top-level scope @ Local: 1

```
1 df2cleancol2 = @pipe df2cleancol |>
2     groupby(_, :participant_num_str) |> # group by participant
3     transform(_, [:resp_arousal_z, :video_scr_z] .=> mean)|>
4     transform(_, [:resp_arousal_z, :resp_arousal_z_mean] => ByRow(-) =>
:resp_arousal_z_cmc, [:video_scr_z, :video_scr_z_mean]=> ByRow(-) =>
:video_scr_z_cmc) |> dropmissing
```

dropnan (generic function with 1 method)

```
1 function dropnan(A)
2     boolmask = (!).(any.(eachrow(isnan.(A))))
3     return A[boolmask,:]
4 end
```

```
dropnanbm (generic function with 1 method)
```

```
1 function dropnanbm(A)
2     boolmask = (!).(any.(eachrow(isnan.(A))))
3     return boolmask
4 end
```

```
UndefVarError: df2cleancol2 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 bmask = dropnanbm(df2cleancol2[:,[:resp_arousal_z_mean,
:resp_arousal_z_cmc,:video_scr_z_mean,:video_scr_z_cmc]])
```

```
UndefVarError: df2cleancol2 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 df2cleancol3 = df2cleancol2[bmask,:]
```

```
UndefVarError: df2cleancol3 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 gdf3 = groupby(df2cleancol3, :participant_num_str);
```

```
UndefVarError: gdf3 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 hpairr = combine(gdf3, :social_phobia => mean, renamecols=false).social_phobia;
```

Zcored Model 1: src to arousal

```
UndefVarError: df2cleancol3 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 mmz1 = fit(LinearMixedModel, @formula(resp_arousal_z ~ video_scr_z_cmc +
video_scr_z_mean + (video_scr_z_cmc|participant_num_str)), df2cleancol3 )
```

```
UndefVarError: mmz1 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 md"#### Zscored Scr to arousal, loglike: $(loglikelihood(mmz1))"
```

```
UndefVarError: mmz1 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 md"#### Zscored Scr to arousal, AIC: $(aic(mgz1))"
```

```
UndefVarError: mmz1 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 mmz1coef_w = cor(ranef(mgz1)[1][2,:], hparr)
```

```
UndefVarError: mmz1 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 mmz1coef_i = cor(ranef(mgz1)[1][1,:], hparr)
```

```
UndefVarError: mmz1 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 scatter(ranef(mgz1)[1][1,:], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(mgz1coef_i)))", xlabel="SCR cluster mean centering MLM raneff  
Intercept", ylabel="social Phobia")
```

```
UndefVarError: mmz1 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 scatter(ranef(mgz1)[1][2,:], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(mgz1coef_w)))", xlabel="SCR cluster mean centering MLM raneff  
Weight", ylabel="social Phobia")
```

Zscored Model 2: arousal to fear

```
UndefVarError: df2cleancol3 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 mmz2 = fit(LinearMixedModel, @formula(resp_fear_z ~ resp_arousal_z_cmc +  
resp_arousal_z_mean + (resp_arousal_z_cmc|participant_num_str)), df2cleancol3)
```

```
UndefVarError: mmz2 not defined
```

```
1. top-level scope @ [ Local: 1 ]
```

```
1 md"#### Zscored Scr to arousal, loglike: $(loglikelihood(mgz2))"
```

UndefVarError: mmz2 not defined

1. top-level scope @ [Local: 1]

```
1 md"#### Zscored Scr to arousal, AIC: $(aic(mmz2))"
```

UndefVarError: mmz2 not defined

1. top-level scope @ [Local: 1]

```
1 mmz2coef_w = cor(ranef(mmz2)[1][2,:], hparr)
```

UndefVarError: mmz2 not defined

1. top-level scope @ [Local: 1]

```
1 mmz2coef_i = cor(ranef(mmz2)[1][1,:], hparr)
```

UndefVarError: mmz2 not defined

1. top-level scope @ [Local: 1]

```
1 scatter(ranef(mmz2)[1][1,:], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(mmz2coef_i)))", xlabel="SCR cluster mean centering MLM raneff  
Intercept", ylabel="social Phobia")
```

UndefVarError: mmz2 not defined

1. top-level scope @ [Local: 1]

```
1 scatter(ranef(mmz2)[1][2,:], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(mmz2coef_w)))", xlabel="SCR cluster mean centering MLM raneff  
Weight", ylabel="social Phobia")
```

Zscored Model 3: scr to fear

```
1 md"## Zscored Model 3: scr to fear"
```

UndefVarError: df2cleancol3 not defined

1. top-level scope @ [Local: 1]

```
1 mmz3 = fit(LinearMixedModel, @formula(resp_fear_z ~ video_scr_z_cmc +  
video_scr_z_mean + (video_scr_z_cmc|participant_num_str)), df2cleancol3)
```

UndefVarError: mmz3 not defined

1. top-level scope @ [Local: 1]

```
1 md"#### Scr to fear, loglike: $(loglikelihood(mmz3))"
```

UndefVarError: mmz3 not defined

1. top-level scope @ Local: 1

```
1 md"#### Scr to fear, AIC: $(aic(mmz3))"
```

UndefVarError: mmz3 not defined

1. top-level scope @ Local: 1

```
1 mmz3coef_w = cor(ranef(mmz3)[1][2,:], hparr)
```

UndefVarError: mmz3 not defined

1. top-level scope @ Local: 1

```
1 mmz3coef_i = cor(ranef(mmz3)[1][1,:], hparr)
```

UndefVarError: mmz3 not defined

1. top-level scope @ Local: 1

```
1 scatter(ranef(mmz3)[1][1,:], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(mmz3coef_i)))", xlabel="SCR cluster mean centering MLM raneff  
Intercept", ylabel="social Phobia")
```

UndefVarError: mmz3 not defined

1. top-level scope @ Local: 1

```
1 scatter(ranef(mmz3)[1][2,:], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(mmz3coef_w)))", xlabel="SCR cluster mean centering MLM raneff  
Weight", ylabel="social Phobia")
```

Zscored Model correlation comparison plots

```
1 md"## Zscored Model correlation comparison plots"
```

UndefVarError: mmz1coef_i not defined

1. top-level scope @ Local: 1

```
1 plotdata_z = [mmz1coef_i,mmz2coef_i,mmz3coef_i];
```

```
1 plotcols_z = ["scr_to_aro_int","aro_to_fear_int", "scr_to_fear_int"];
```

UndefVarError: plotdata_z not defined

1. top-level scope @ [Local: 1]

```
1 plot(bar(plotcols_z, plotdata_z), title="Intercept term correlations to trait phobia", xlabel="Models", ylabel="Corr coef")
```

UndefVarError: mmz1coef_w not defined

1. top-level scope @ [Local: 1]

```
1 plotdata_z1 = [mmz1coef_w,mmz2coef_w,mmz3coef_w];
```

```
1 plotcols_z1 = ["scr_to_aro_beta","aro_to_fear_beta", "scr_to_fear_beta"];
```

UndefVarError: plotdata_z1 not defined

1. top-level scope @ [Local: 1]

```
1 plot(bar(plotcols_z1, plotdata_z1), title="Beta/Weight term correlations to trait phobia", xlabel="Models", ylabel="Corr coef")
```

UndefVarError: mmz1 not defined

1. top-level scope @ [Local: 1]

```
1 models2 = [mmz1,mmz2,mmz3];
```

```
1 plotcols_z3 = ["scr_to_aro","aro_to_fear", "scr_to_fear"];
```

UndefVarError: models2 not defined

1. top-level scope @ [Local: 1]

```
1 plot(bar(plotcols_z3, models2 .|> loglikelihood), title="Log Likelihood model fit (higher = better)", xlabel="Models", ylabel="Log Likelihood")
```

UndefVarError: models2 not defined

1. top-level scope @ [Local: 1]

```
1 plot(bar(plotcols_z3, models2 .|> aic), title="AIC model fit (lower = better)", xlabel="Models", ylabel="AIC")
```