

# Modeling Notebook

```
1 md"# Modeling Notebook"
```

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

```
cond_num = 3
```

```
1 cond_num = 3
```

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

```
► ["107", "107", "107", "107", "107", "107", "107", "107", "107", "107", "107", "107", "107", "107", "107"]
```

```
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_\
<b>1</b>	0	0	"spider_high_5.m4v"	3	2	2
<b>2</b>	1	1	"spider_low_3.m4v"	3	1	1
<b>3</b>	2	2	"spider_low_6.m4v"	3	1	1
<b>4</b>	7	7	"spider_high_6.mov"	3	2	2
<b>5</b>	13	13	"spider_low_4.m4v"	3	1	2
<b>6</b>	17	17	"spider_high_1.mov"	3	2	2
<b>7</b>	18	18	"spider_high_4.m4v"	3	2	1
<b>8</b>	19	19	"spider_low_1.m4v"	3	1	1
<b>9</b>	28	28	"spider_low_2.mov"	3	1	2
<b>10</b>	29	29	"spider_high_3.mov"	3	2	1
: more						
<b>1234</b>	3788	3789	"spider_low_5.mov"	3	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_\
<b>1</b>	0	0	"spider_high_5.m4v"	3	2	2
<b>2</b>	1	1	"spider_low_3.m4v"	3	1	1
<b>3</b>	2	2	"spider_low_6.m4v"	3	1	1
<b>4</b>	7	7	"spider_high_6.mov"	3	2	2
<b>5</b>	13	13	"spider_low_4.m4v"	3	1	2
<b>6</b>	17	17	"spider_high_1.mov"	3	2	2
<b>7</b>	18	18	"spider_high_4.m4v"	3	2	1
<b>8</b>	19	19	"spider_low_1.m4v"	3	1	1
<b>9</b>	28	28	"spider_low_2.mov"	3	1	2
<b>10</b>	29	29	"spider_high_3.mov"	3	2	1
: more						
<b>1234</b>	3788	3789	"spider_low_5.mov"	3	1	1

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

## Correlation Table

```
df3_extra =
```

---

	participant_num	spider_phobia_mean_miss	video_scr_mean_miss	video_hp_mean_
<b>1</b>	107	1.27778	0.0476944	0.927861
<b>2</b>	108	5.0	0.0416667	0.786829
<b>3</b>	109	1.77778	0.0664167	0.863139
<b>4</b>	110	1.66667	0.0737647	0.831937
<b>5</b>	112	2.88889	0.0319444	1.06551
<b>6</b>	114	4.66667	NaN	NaN
<b>7</b>	115	2.38889	0.0575833	0.813192
<b>8</b>	118	1.11111	0.0490278	0.9876
<b>9</b>	119	1.44444	0.00288889	0.675694
<b>10</b>	120	1.0	0.0414444	0.938516
: more				
<b>105</b>	232	3.72222	0.0296364	0.992788

---

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

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

```
cordfextra =
```

	spider_phobia	video_scr	video_hp	resp_arousal	resp_fear	resp_valence
<b>1</b>	1.0	0.0848113	-0.144882	0.111721	0.387516	-0.335958
<b>2</b>	0.0848113	1.0	-0.183235	0.142326	0.0258892	-0.0571948
<b>3</b>	-0.144882	-0.183235	1.0	0.0407244	-0.154579	0.141462
<b>4</b>	0.111721	0.142326	0.0407244	1.0	0.378368	-0.0900944
<b>5</b>	0.387516	0.0258892	-0.154579	0.378368	1.0	-0.61302
<b>6</b>	-0.335958	-0.0571948	0.141462	-0.0900944	-0.61302	1.0

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

```
1 CSV.write("phobiacordfspider.csv", cordfextra)
```

## Test Area

	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	7	7	"spider_high_6.mov"	3	2	2
5	13	13	"spider_low_4.m4v"	3	1	2
6	17	17	"spider_high_1.mov"	3	2	2
7	18	18	"spider_high_4.m4v"	3	2	1
8	19	19	"spider_low_1.m4v"	3	1	1
9	28	28	"spider_low_2.mov"	3	1	2
10	29	29	"spider_high_3.mov"	3	2	1
⋮ more						
1234	3788	3789	"spider_low_5.mov"	3	1	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 "spider_phobia",
22 "participant_num_str"];
```

```
dfcleancol =
```

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_a
1	"spider_high_5.m4v"	107	0.248	3.22	0.125
2	"spider_low_3.m4v"	107	0.318	2.636	0.141
3	"spider_low_6.m4v"	107	0.347	3.571	0.227
4	"spider_high_6.mov"	107	0.225	2.086	0.126
5	"spider_low_4.m4v"	107	0.052	1.485	0.288
6	"spider_high_1.mov"	107	0.267	2.57	0.218
7	"spider_high_4.m4v"	107	0.311	5.623	0.269
8	"spider_low_1.m4v"	107	0.306	7.525	0.31
9	"spider_low_2.mov"	107	0.044	1.435	0.06
10	"spider_high_3.mov"	107	0.141	1.902	0.102
: more					
1234	"spider_low_5.mov"	232	0.165	32.386	0.091

```
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 "spider_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	"spider_low_6.m4v"	224	0.498	0.718	0.5
2	"spider_low_3.m4v"	224	0.5	1.335	0.5
3	"spider_high_3.mov"	224	0.5	0.417	0.5
4	"spider_low_4.m4v"	224	0.5	0.468	0.5
5	"spider_low_1.m4v"	224	0.5	0.467	0.5
6	"spider_high_4.m4v"	224	0.497	2.003	0.5
7	"spider_high_6.mov"	224	0.5	1.135	0.5
8	"spider_low_2.mov"	224	0.5	1.051	0.5
9	"spider_high_5.m4v"	224	0.508	1.352	0.471
10	"spider_high_2.m4v"	224	0.5	1.302	0.5
11	"spider_low_5.mov"	224	0.5	2.219	0.5

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

```
▶ [0.051, 0.051, 0.051, 0.054, 0.0, 0.0, 0.154, 0.0, 0.051, 0.105, 0.105, 0.0, 0.051, 0.0]
```

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

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

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

```
0.09255042746275481
```

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

```
0.1888482502742108
```

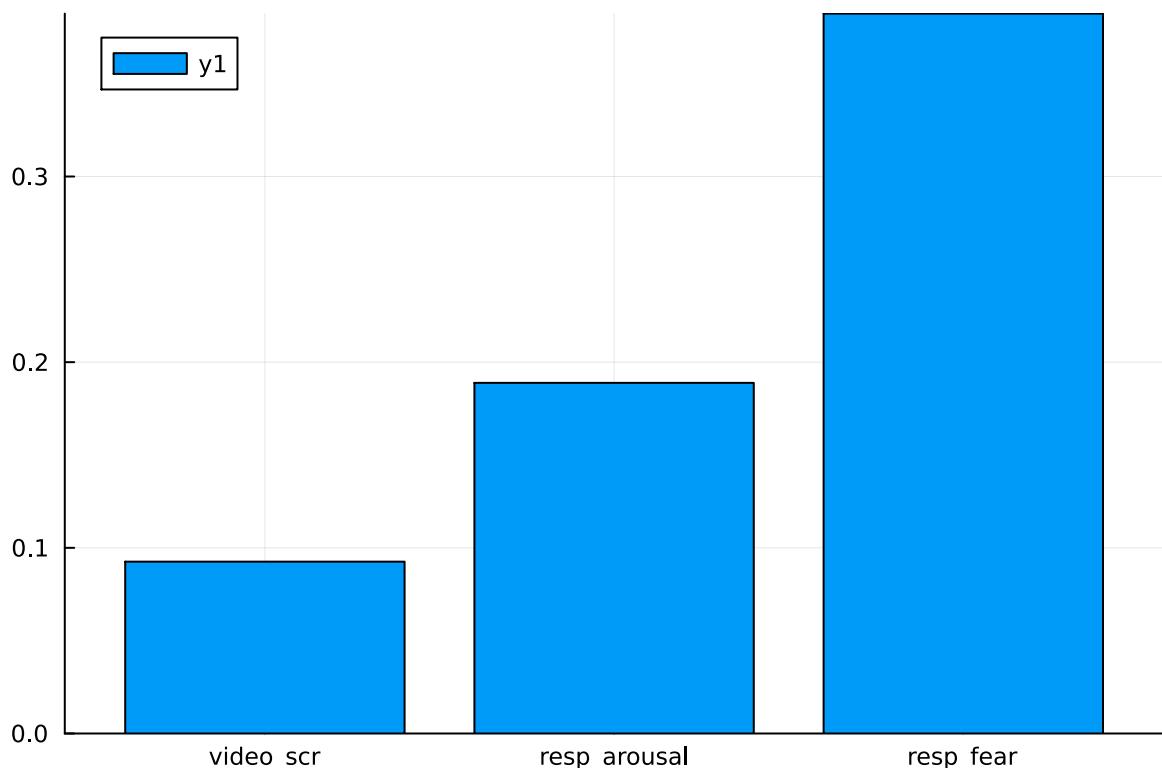
```
1 comparevar("resp_arousal")
```

```
0.387738259886215
```

```
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)	0.0213729	0.00250989	8.52	<1e-16	0.0164488	0.0262971
resp_arousal	0.0234106	0.0044209	5.30	<1e-06	0.0147373	0.0320839

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

```
0.021372941811899207
```

```
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	resp_current_ar
	String31	Int64	Float64	Float64	Float64
1	spider_high_5.m4v	107	0.248	3.22	
2	spider_low_3.m4v	107	0.318	2.636	
3	spider_low_6.m4v	107	0.347	3.571	
4	spider_high_6.mov	107	0.225	2.086	
5	spider_low_4.m4v	107	0.052	1.485	
6	spider_high_1.mov	107	0.267	2.57	
7	spider_high_4.m4v	107	0.311	5.623	
8	spider_low_1.m4v	107	0.306	7.525	
9	spider_low_2.mov	107	0.044	1.435	
10	spider_high_3.mov	107	0.141	1.902	
11	spider_high_2.m4v	107	0.183	3.504	
12	spider_low_5.mov	107	0.236	2.269	

:

Last Group (11 rows): participant\_num\_str = "232"

Row	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_ar
	String31	Int64	Float64	Float64	Float64
1	spider_low_2.mov	232	0.396	2.136	
2	spider_low_3.m4v	232	0.245	2.637	
3	spider_high_5.m4v	232	0.244	3.07	
4	spider_high_4.m4v	232	0.218	4.672	
5	spider_high_3.mov	232	0.287	2.486	
6	spider_low_4.m4v	232	0.269	4.238	
7	spider_low_6.m4v	232	0.199	3.721	
8	spider_high_2.m4v	232	0.213	3.254	
9	spider_high_1.mov	232	0.355	2.937	
10	spider_high_6.mov	232	0.335	5.089	
11	spider_low_5.mov	232	0.165	32.386	

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

---

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_anx
1	"spider_low_1.m4v"	114	0.911	3.087	0.761
2	"spider_low_5.mov"	114	0.631	2.987	0.5
3	"spider_high_1.mov"	114	0.733	3.07	0.623
4	"spider_high_6.mov"	114	0.753	1.219	0.822
5	"spider_low_4.m4v"	114	0.835	1.519	0.899
6	"spider_high_3.mov"	114	0.5	1.552	0.5
7	"spider_low_2.mov"	114	0.5	0.484	0.5
8	"spider_high_2.m4v"	114	0.627	1.018	0.642
9	"spider_low_3.m4v"	114	0.5	1.802	0.5
10	"spider_high_5.m4v"	114	0.66	2.236	0.675
11	"spider_high_4.m4v"	114	0.425	0.851	0.379
12	"spider_low_6.m4v"	114	0.5	2.57	0.5

---

```
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 "spider_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

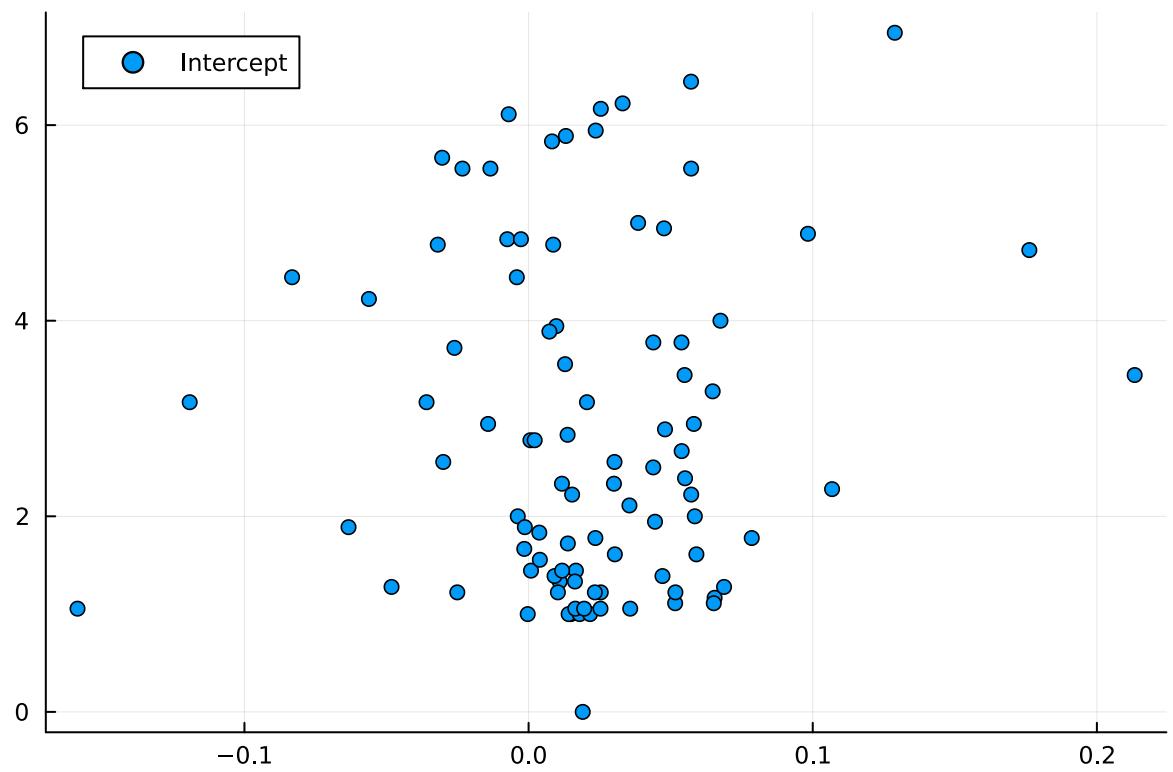
`df2_subjlv1 =`

	participant_num_str	video_scr_resp_arousal	spider_phobia
<b>1</b>	"107"	► (-0.0482363, 0.138281, 23.302)	1.27778
<b>2</b>	"108"	► (0.038541, -0.0206362, 27.1027)	5.0
<b>3</b>	"109"	► (0.0785308, -0.0317953, 22.2598)	1.77778
<b>4</b>	"110"	► (-0.00152242, 0.0777193, 20.6283)	1.66667
<b>5</b>	"112"	► (0.0480059, -0.0533271, 28.8427)	2.88889
<b>6</b>	"115"	► (0.0550599, 0.00189484, 26.8397)	2.38889
<b>7</b>	"118"	► (0.0515734, -0.0696129, 24.5181)	1.11111
<b>8</b>	"119"	► (0.000848967, 0.00870702, 33.7626)	1.44444
<b>9</b>	"120"	► (0.0148595, 0.0528454, 17.7078)	1.0
<b>10</b>	"122"	► (-0.0300345, 0.11789, 22.6896)	2.55556
: more			
<b>94</b>	"232"	► (-0.0260921, 0.0773472, 31.7896)	3.72222

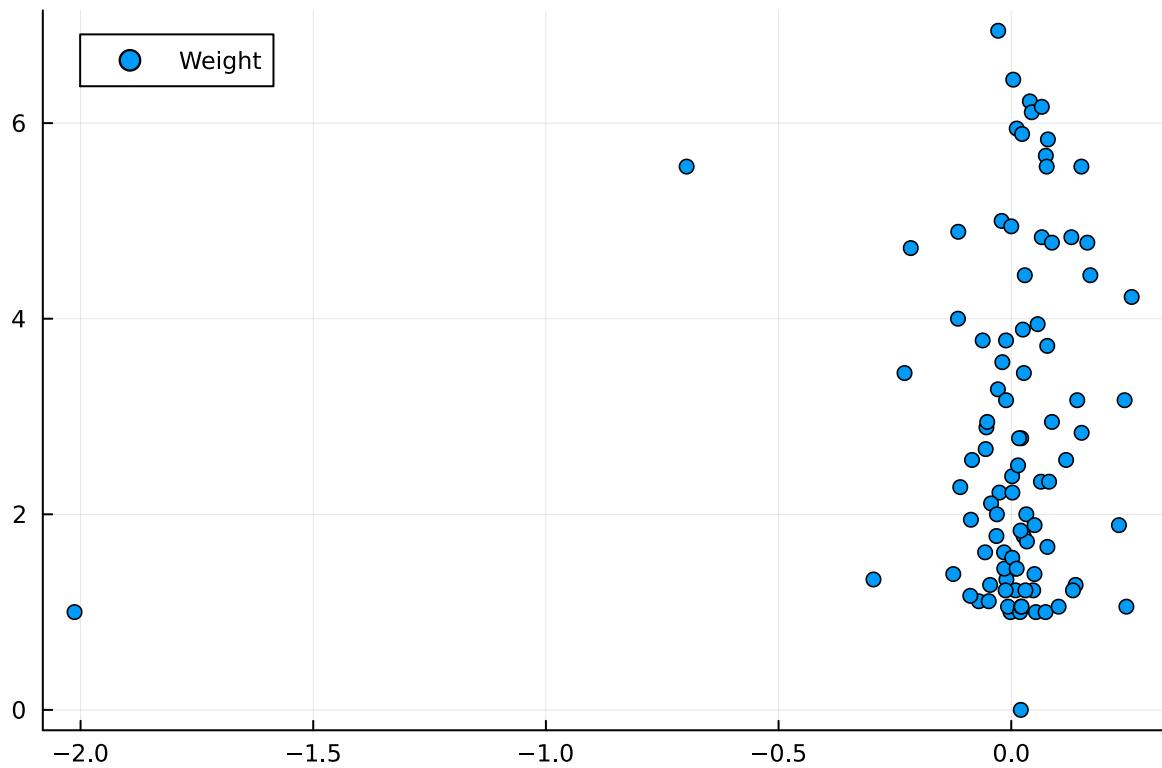
```

1 df2_subjlv1 = combine(gdf, [:video_scr, :resp_arousal] => meantest,
:spider_phobia => mean, renamecols=false) |> dropmissing

```



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



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

```
scr_to_aro_int = 0.0711324375029642
```

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

```
scr_to_aro_coef = 0.08945677324349646
```

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

```
▶ [23.302, 27.1027, 22.2598, 20.6283, 28.8427, 26.8397, 24.5181, 33.7626, 17.7078, 22.68
```

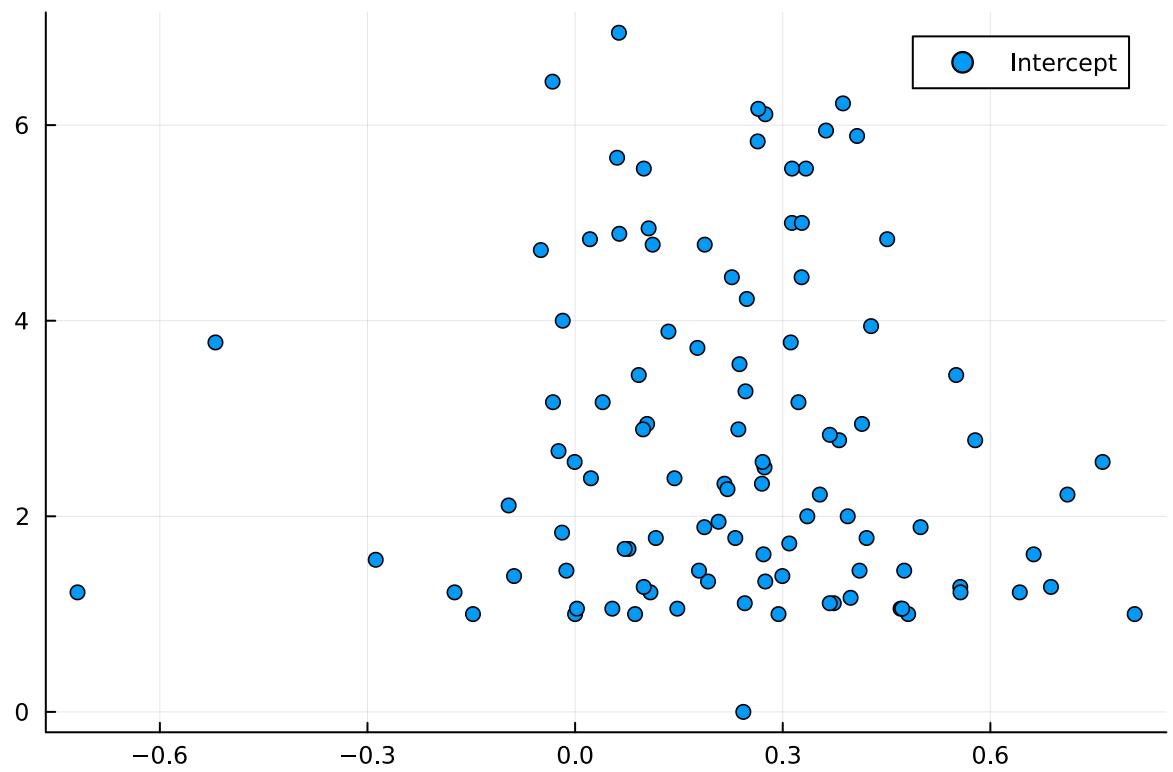
```
1 df2_subjlvl.video_scr_resp_arousal .|> x -> x[3]
```

## Fitting Arousal to Fear Response

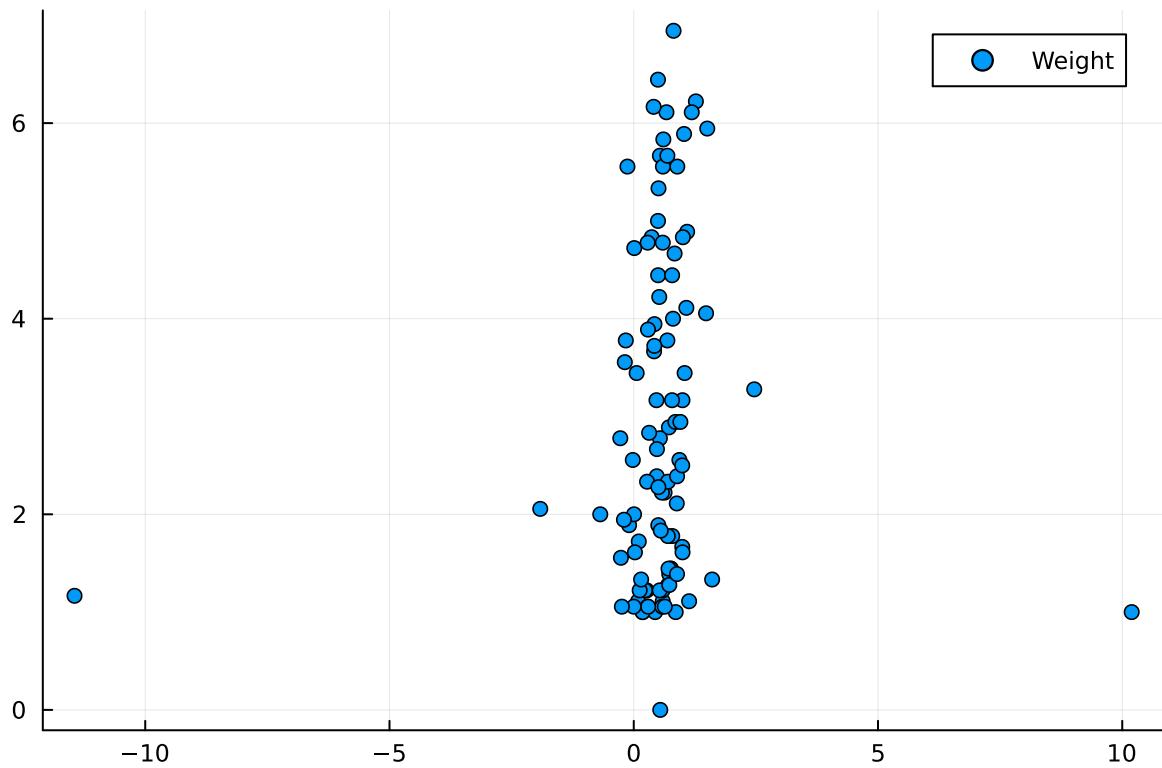
```
df2_subjlvl_2 =
```

	participant_num_str	resp_arousal_resp_fear	spider_phobia
<b>1</b>	"107"	► (0.556501, 0.705588, 2.90634)	1.27778
<b>2</b>	"108"	► (0.31325, 0.496559, 3.0943)	5.0
<b>3</b>	"109"	► (0.231445, 0.787861, 3.93494)	1.77778
<b>4</b>	"110"	► (0.0771862, 0.993122, 5.8646)	1.66667
<b>5</b>	"112"	► (0.235863, 0.719449, 18.0397)	2.88889
<b>6</b>	"114"	► (0.143615, 0.836156, 11.6865)	4.66667
<b>7</b>	"115"	► (0.373722, 0.471398, 9.35221)	2.38889
<b>8</b>	"118"	► (0.475503, 0.59271, 8.20731)	1.11111
<b>9</b>	"119"	► (0.0866379, 0.760413, 10.5042)	1.44444
<b>10</b>	"120"	► (0.762186, 0.182778, -0.285981)	1.0
: more			
<b>104</b>	"232"	► (0.270913, 0.420928, 13.2778)	3.72222

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



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



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

```
aro_to_fear_int = -0.2371654869711222
```

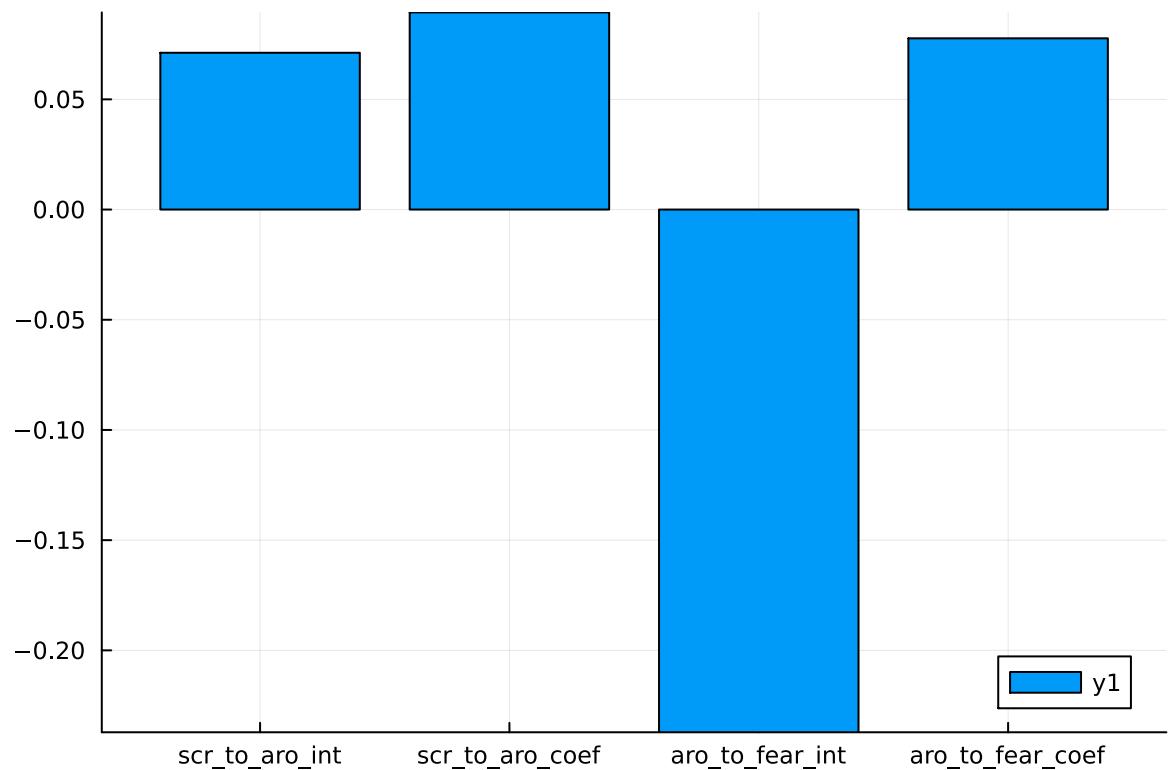
```
1 aro_to_fear_int = cor(df2_subjlv2.resp_arousal_resp_fear .|> x -> x[1],
df2_subjlv2.spider_phobia)
```

```
aro_to_fear_coef = 0.07768884455035523
```

```
1 aro_to_fear_coef = cor(df2_subjlv2.resp_arousal_resp_fear .|> x -> x[2],
df2_subjlv2.spider_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, :spider_phobia => mean, renamecols=false);
```

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_a
<b>1</b>	"spider_high_5.m4v"	107	0.248	3.22	0.125
<b>2</b>	"spider_low_3.m4v"	107	0.318	2.636	0.141
<b>3</b>	"spider_low_6.m4v"	107	0.347	3.571	0.227
<b>4</b>	"spider_high_6.mov"	107	0.225	2.086	0.126
<b>5</b>	"spider_low_4.m4v"	107	0.052	1.485	0.288
<b>6</b>	"spider_high_1.mov"	107	0.267	2.57	0.218
<b>7</b>	"spider_high_4.m4v"	107	0.311	5.623	0.269
<b>8</b>	"spider_low_1.m4v"	107	0.306	7.525	0.31
<b>9</b>	"spider_low_2.mov"	107	0.044	1.435	0.06
<b>10</b>	"spider_high_3.mov"	107	0.141	1.902	0.102
: more					
<b>1234</b>	"spider_low_5.mov"	232	0.165	32.386	0.091

1 dfcleancol

mm2 =

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.4734	0.0208	22.74	<1e-99	0.1987
video_scr	0.4928	0.1472	3.35	0.0008	
Residual	0.2033				

```
Minimizing 2      Time: 0:00:00 (73.50 ms/it)
objective: -123.38510489146381
```

```
Minimizing 13      Time: 0:00:00 (16.08 ms/it)
```

	Column	Variance	Std.Dev
participant_num_str	(Intercept)	0.039501	0.198748
Residual		0.041330	0.203298

```
1 VarCorr(mm2)
```

► [1×105 Matrix{Float64}]:  
0.20667 0.122988 0.188494 -0.0405216 0.16391 ... 0.272974 0.268069 -0.0705926  
1 ranef(mm2)

0.03660481535245002

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

	participant_num_str	(Intercept)
1	"107"	0.20667
2	"108"	0.122988
3	"109"	0.188494
4	"110"	-0.0405216
5	"112"	0.16391
6	"114"	0.138415
7	"115"	0.0763175
8	"118"	0.0079575
9	"119"	-0.060577
10	"120"	0.256636
:	more	
105	"232"	-0.0705926

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

## Mixed Models

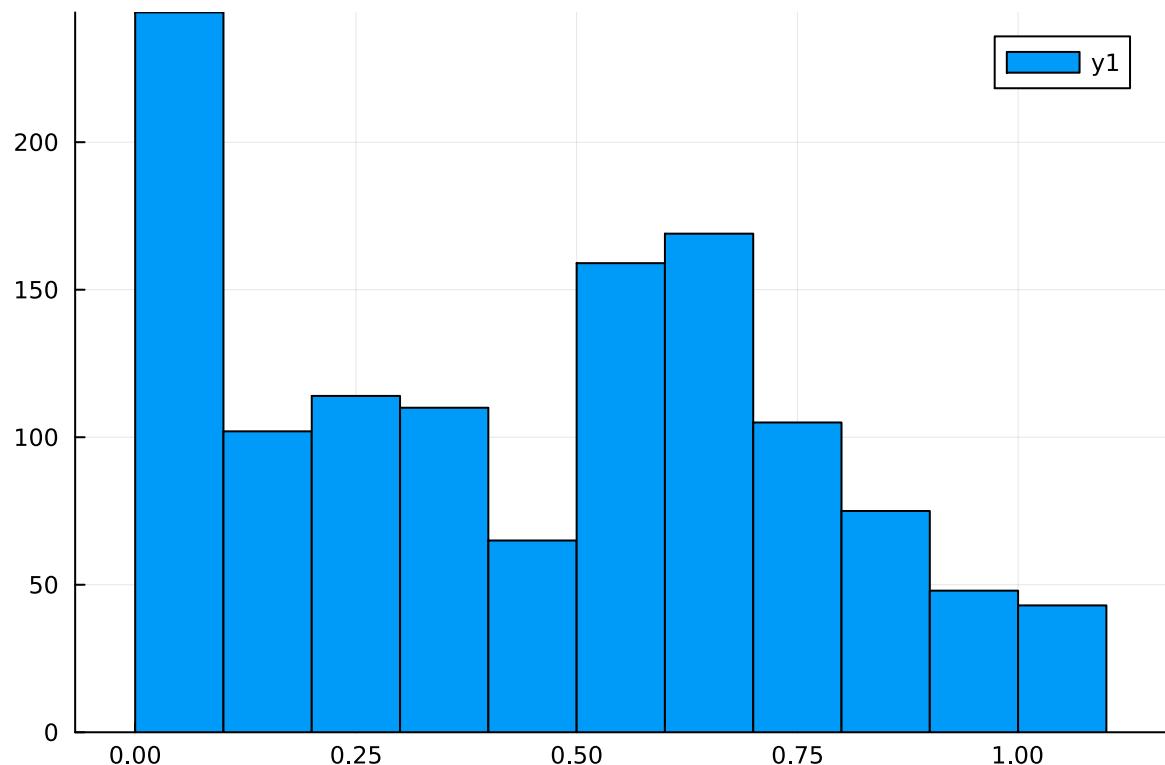
```
1 md"## Mixed Models"
```

```
dfcleancol2 =
```

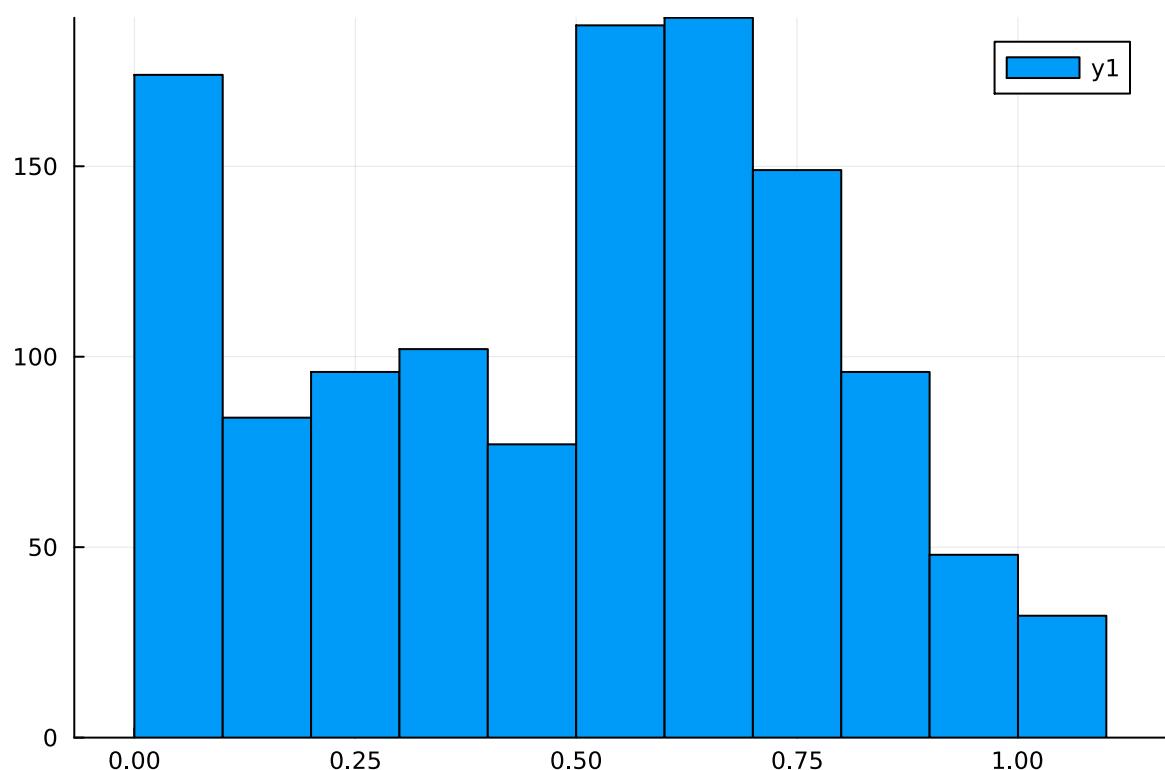
	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_a
1	"spider_high_5.m4v"	107	0.248	3.22	0.125
2	"spider_low_3.m4v"	107	0.318	2.636	0.141
3	"spider_low_6.m4v"	107	0.347	3.571	0.227
4	"spider_high_6.mov"	107	0.225	2.086	0.126
5	"spider_low_4.m4v"	107	0.052	1.485	0.288
6	"spider_high_1.mov"	107	0.267	2.57	0.218
7	"spider_high_4.m4v"	107	0.311	5.623	0.269
8	"spider_low_1.m4v"	107	0.306	7.525	0.31
9	"spider_low_2.mov"	107	0.044	1.435	0.06
10	"spider_high_3.mov"	107	0.141	1.902	0.102
: more					
1234	"spider_low_5.mov"	232	0.165	32.386	0.091

```
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: scr to arousal

```
mm3 =
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.4153	0.0341	12.19	<1e-33	0.1932
video_scr_cmc	0.4128	0.1586	2.60	0.0093	0.4652
video_scr_mean	2.2638	0.8460	2.68	0.0075	
Residual	0.2024				

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

Scr to arousal, loglike: 88.47032504285946

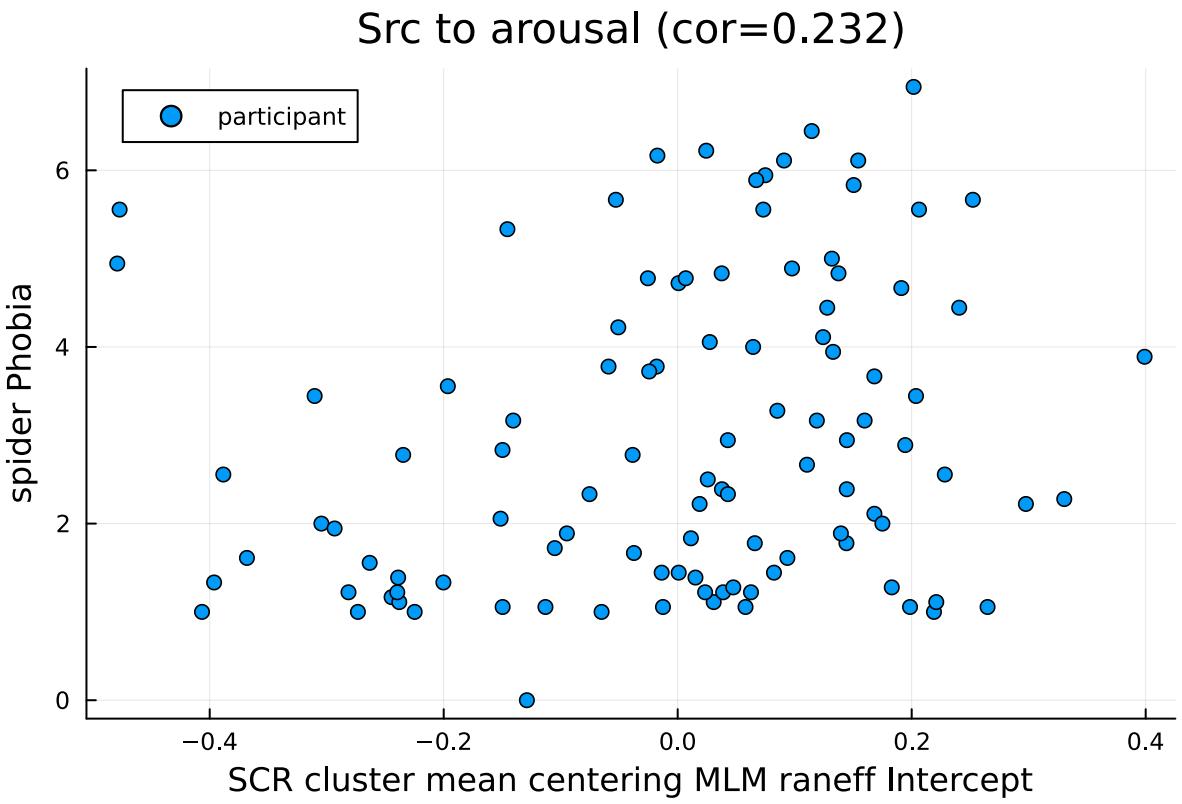
```
1 md"#### Scr to arousal, loglike: $(loglikelihood(mm3))"
```

Scr to arousal, AIC: -162.94065008571891

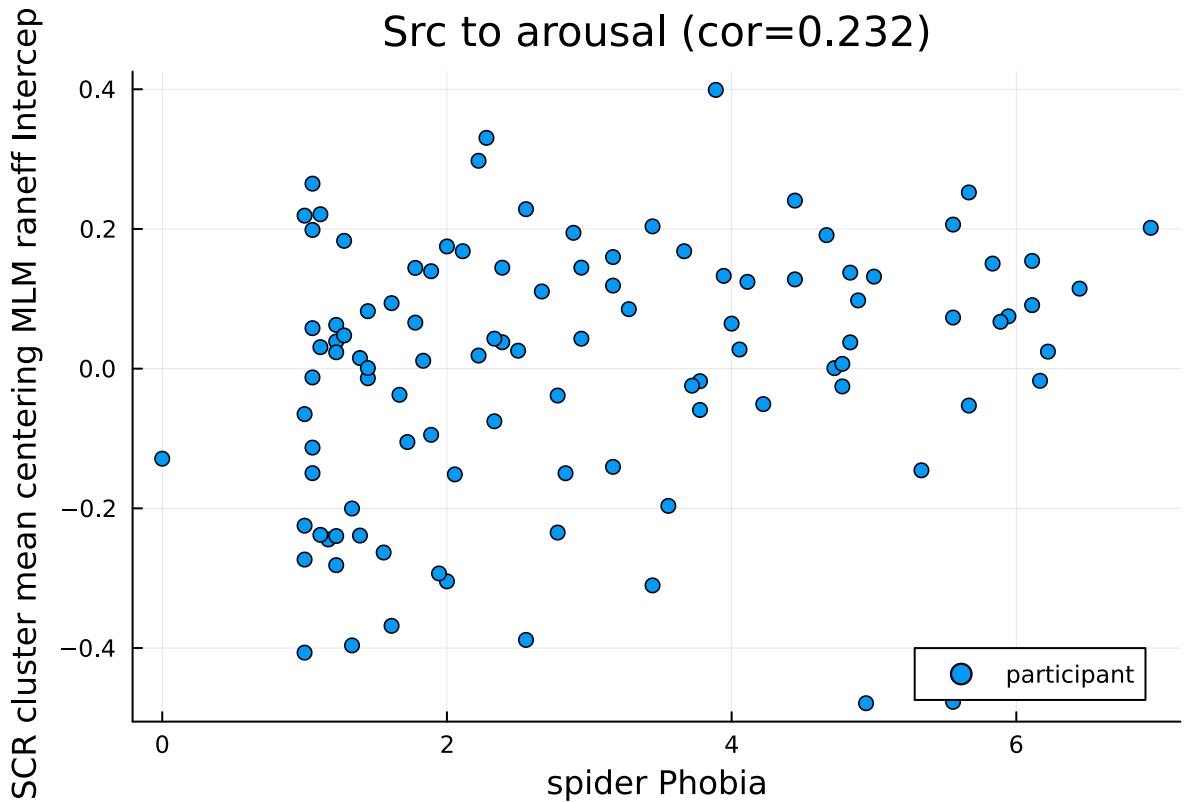
```
1 md"#### Scr to arousal, AIC: $(aic(mm3))"
```

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

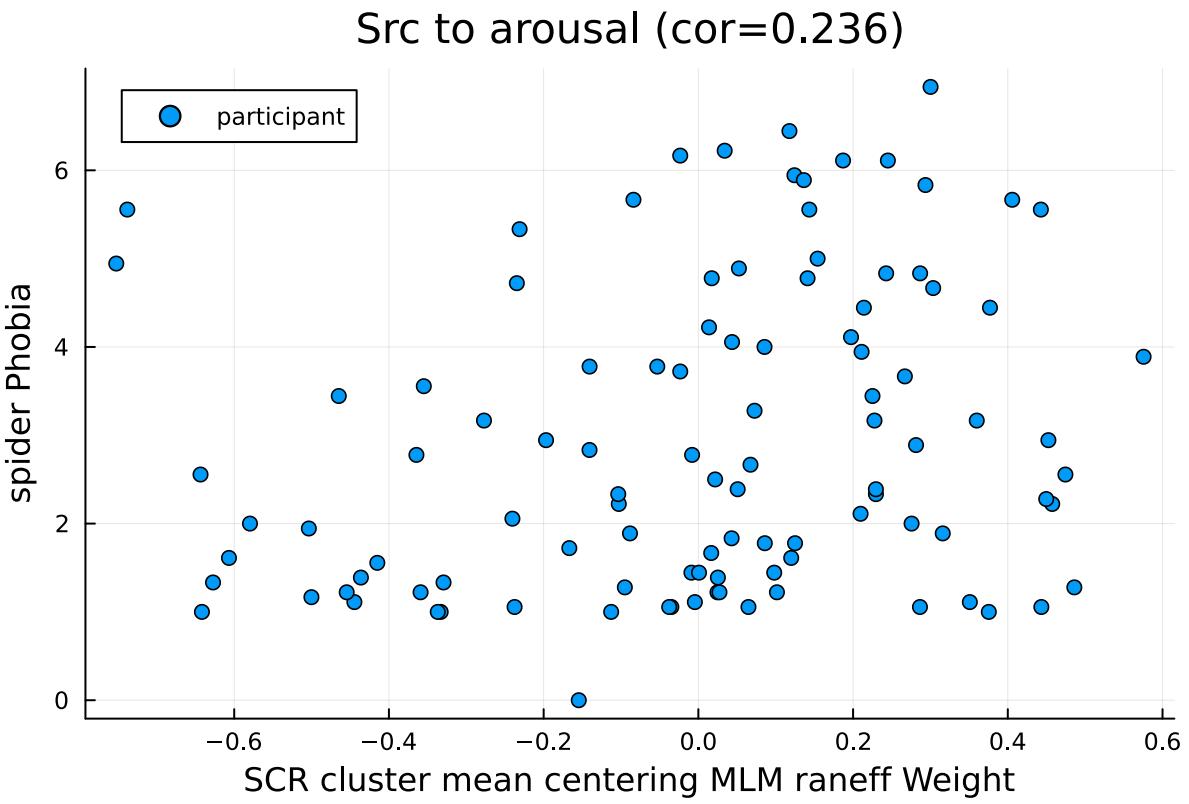
```
1 mm3coef_w = cor(ranef(mm3)[1][2,:], df2_subjlvl_hp.spider_phobia);
```



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



```
1 scatter( df2_subjlv_hp.spider_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="spider Phobia")
```



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

## Model 2: arousal to fear

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

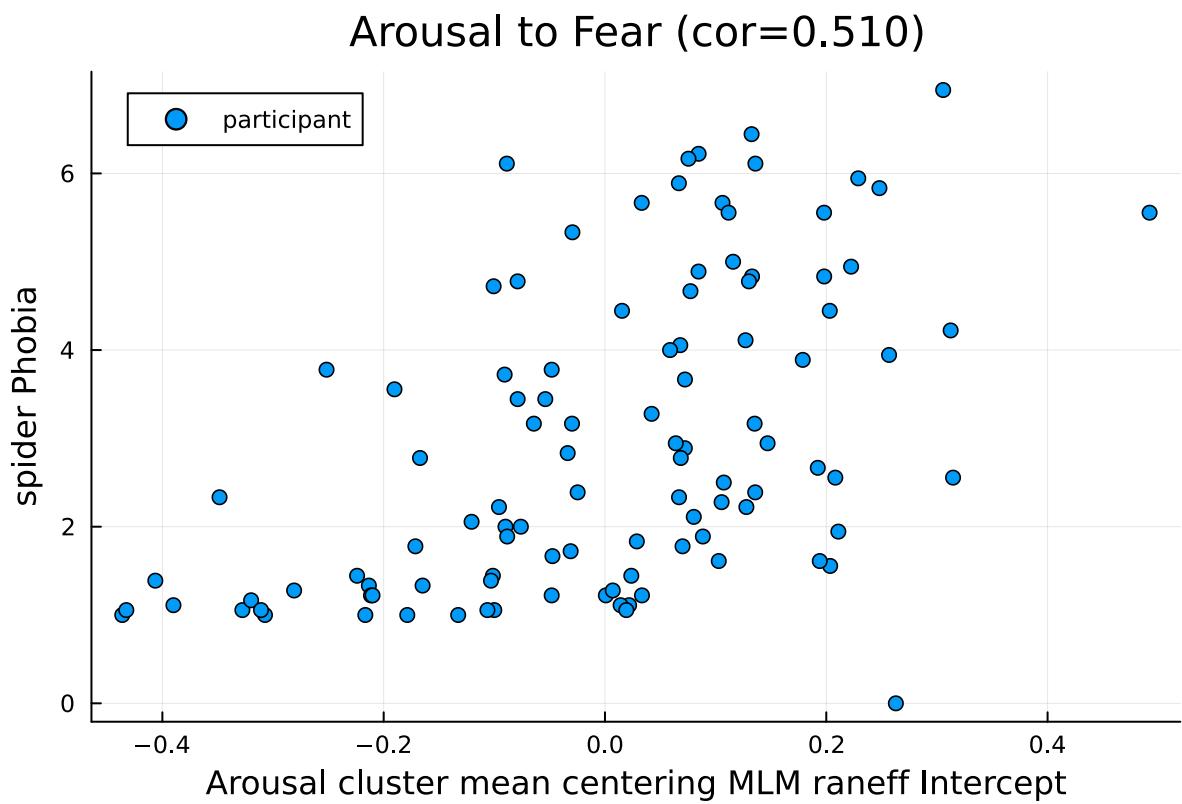
```
mm4 =
```

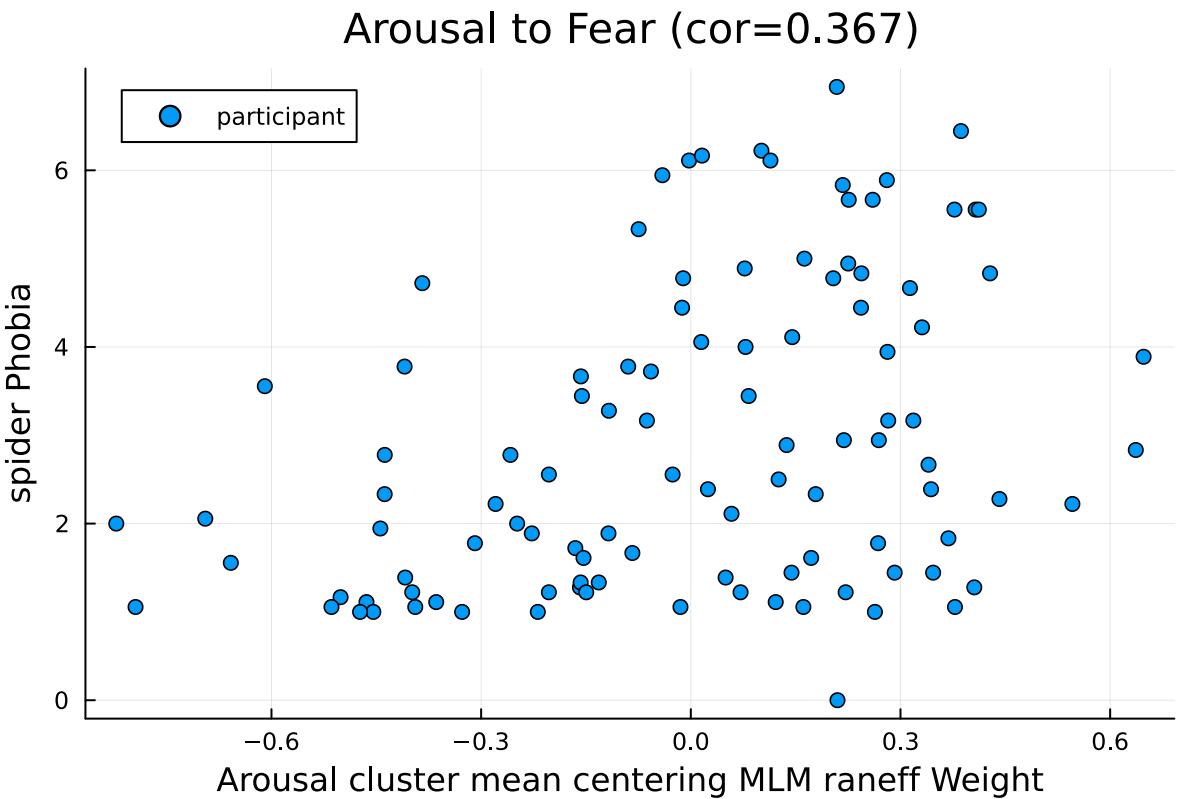
	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.2320	0.0454	5.11	<1e-06	0.1862
resp_arousal_cmc	0.5848	0.0492	11.90	<1e-31	0.3953
resp_arousal_mean	0.4240	0.0843	5.03	<1e-06	
Residual	0.1660				

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

Arousal to fear, loglike: 270.56286405981905

**Arousal to fear, AIC: -527.1257281196381**





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

## Model 3: scr to fear

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

```
mm6 =
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.3548	0.0365	9.71	<1e-21	0.2057
video_scr_cmc	0.5308	0.1613	3.29	0.0010	0.2292
video_scr_mean	2.5849	0.9101	2.84	0.0045	
Residual	0.2162				

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

Scr to fear, loglike: 9.393903405256196

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

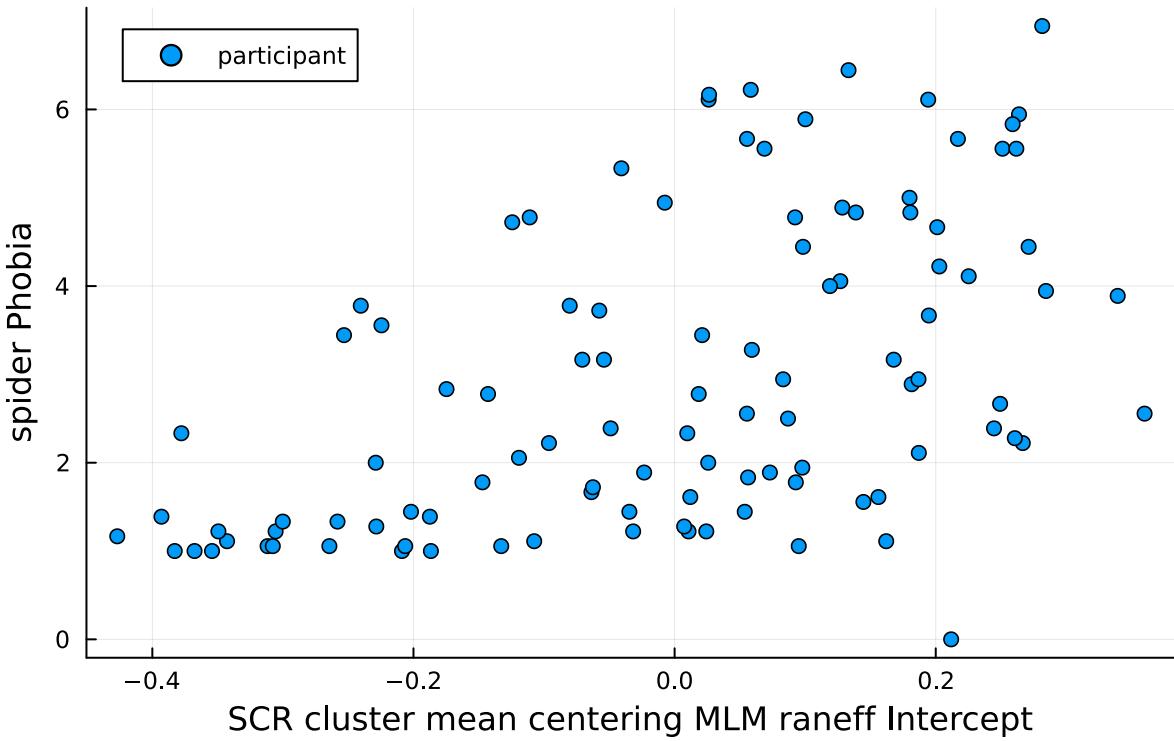
## Scr to fear, AIC: -4.787806810512393

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

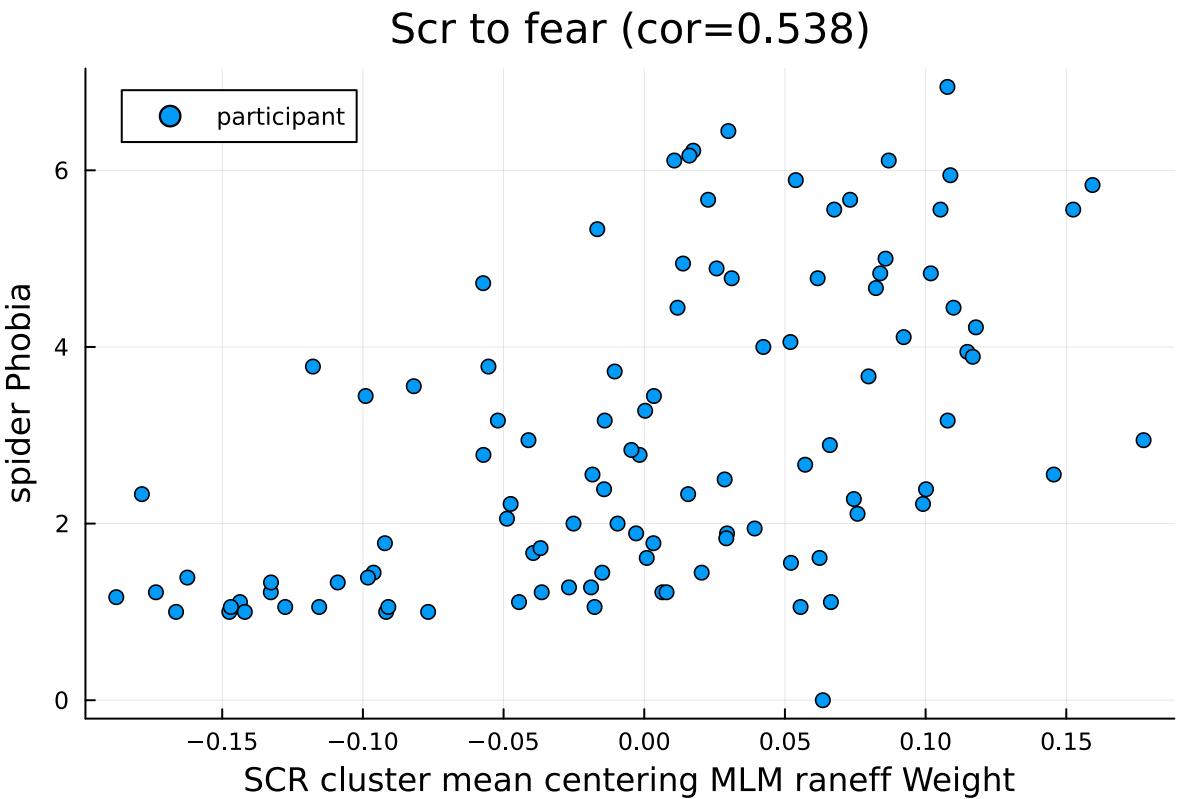
```
1 mm6coef_i = cor(ranef(mm6)[1][1,:], df2_subjlvl_hp.spider_phobia);
```

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

Scr to fear (cor=0.511)



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



```
1 scatter(ranef(mm6)[1][2,:], df2_subjlvl_hp.spider_phobia, label="participant",
  title="Scr to fear (cor=$(get_trunc(mm6$coef_w)))", xlabel="SCR cluster mean
  centering MLM raneff Weight", ylabel="spider Phobia")
```

## Model 4: scr to valence

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

```
mm12 =
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.3638	0.0270	13.49	<1e-40	0.1478
video_scr_cmc	-0.5005	0.1448	-3.46	0.0005	0.0488
video_scr_mean	-1.8716	0.6719	-2.79	0.0053	
Residual	0.1969				

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

Scr to arousal, loglike: 147.71981836596245

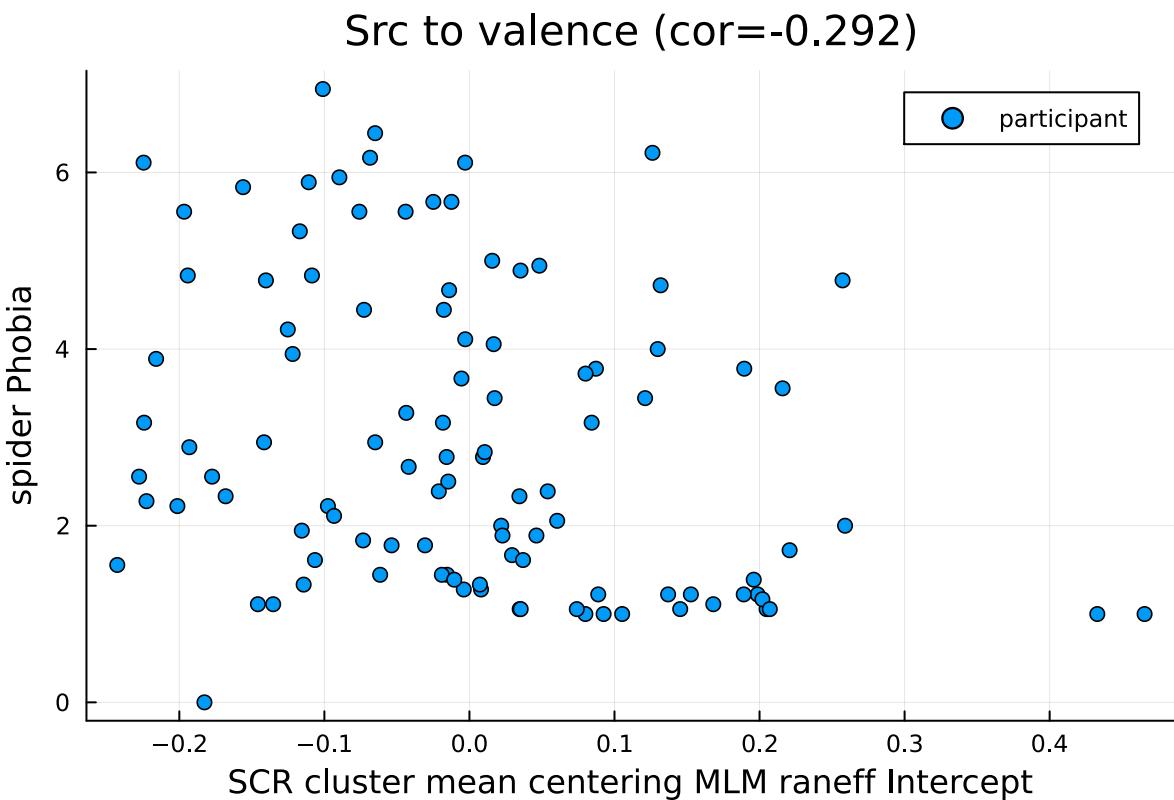
```
1 md"#### Scr to arousal, loglike: $(loglikelihood(mm12))"
```

## Scr to arousal, AIC: -281.4396367319249

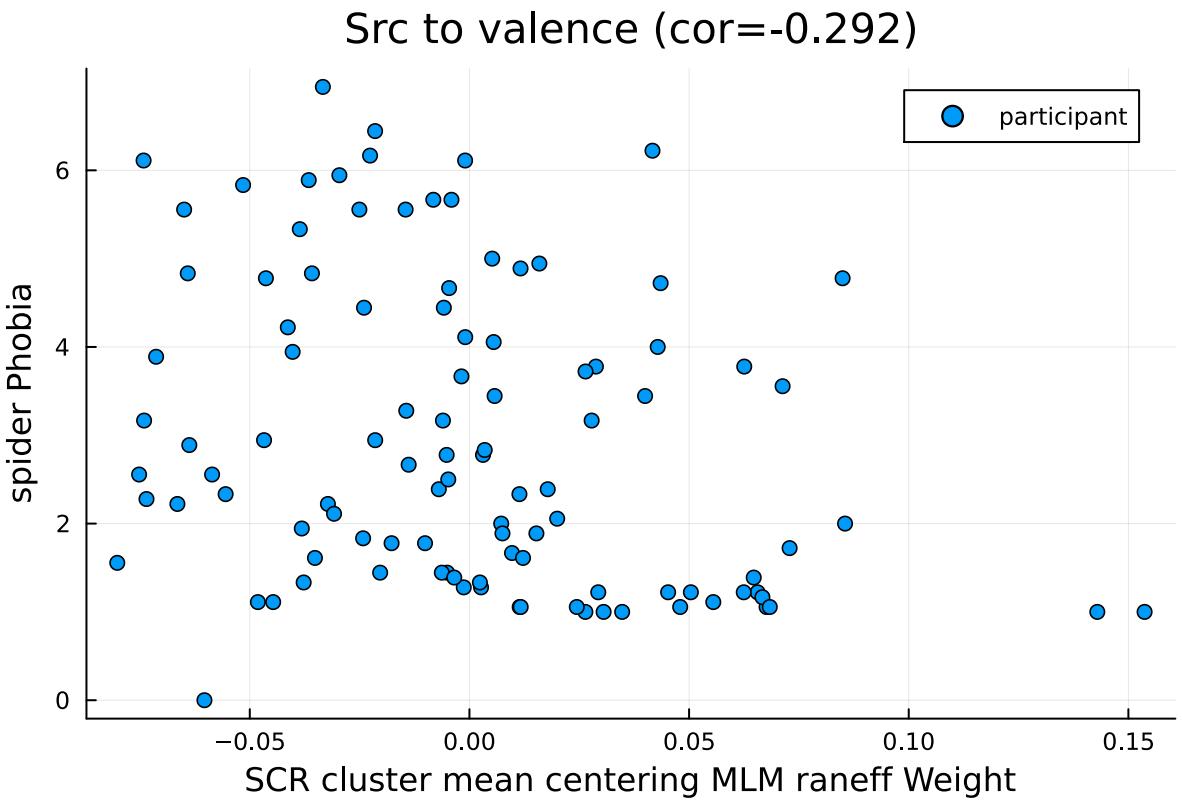
```
1 md"#### Scr to arousal, AIC: $(aic(mm12))"
```

```
1 mm12coef_i = cor(ranef(mm12)[1][1,:], df2_subjlvl_hp.spider_phobia);
```

```
1 mm12coef_w = cor(ranef(mm12)[1][2,:], df2_subjlvl_hp.spider_phobia);
```



```
1 scatter(ranef(mm12)[1][1,:], df2_subjlvl_hp.spider_phobia, label="participant",  
title="Src to valence (cor=$get_trunc(mm12coef_i))", xlabel="SCR cluster mean  
centering MLM raneff Intercept", ylabel="spider Phobia")
```



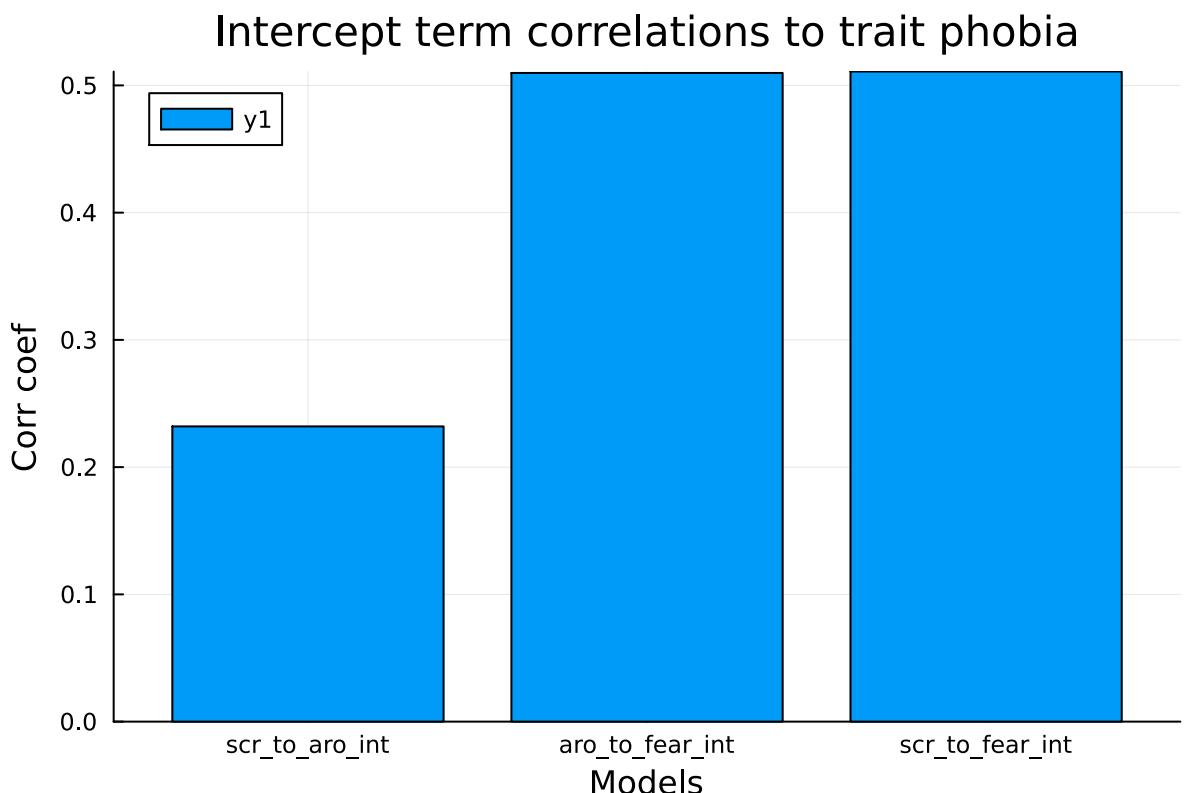
```
1 scatter(ranef(mm12)[1][2,:], df2_subjlvl_hp.spider_phobia, label="participant",
  title="Src to valence (cor=$(get_trunc(mm12coef_w)))", xlabel="SCR cluster mean
  centering MLM raneff Weight", ylabel="spider Phobia")
```

## Model correlation comparison plots

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

```
1 plotdata3 = [mm3coef_i,mm4coef_i,mm6coef_i];
```

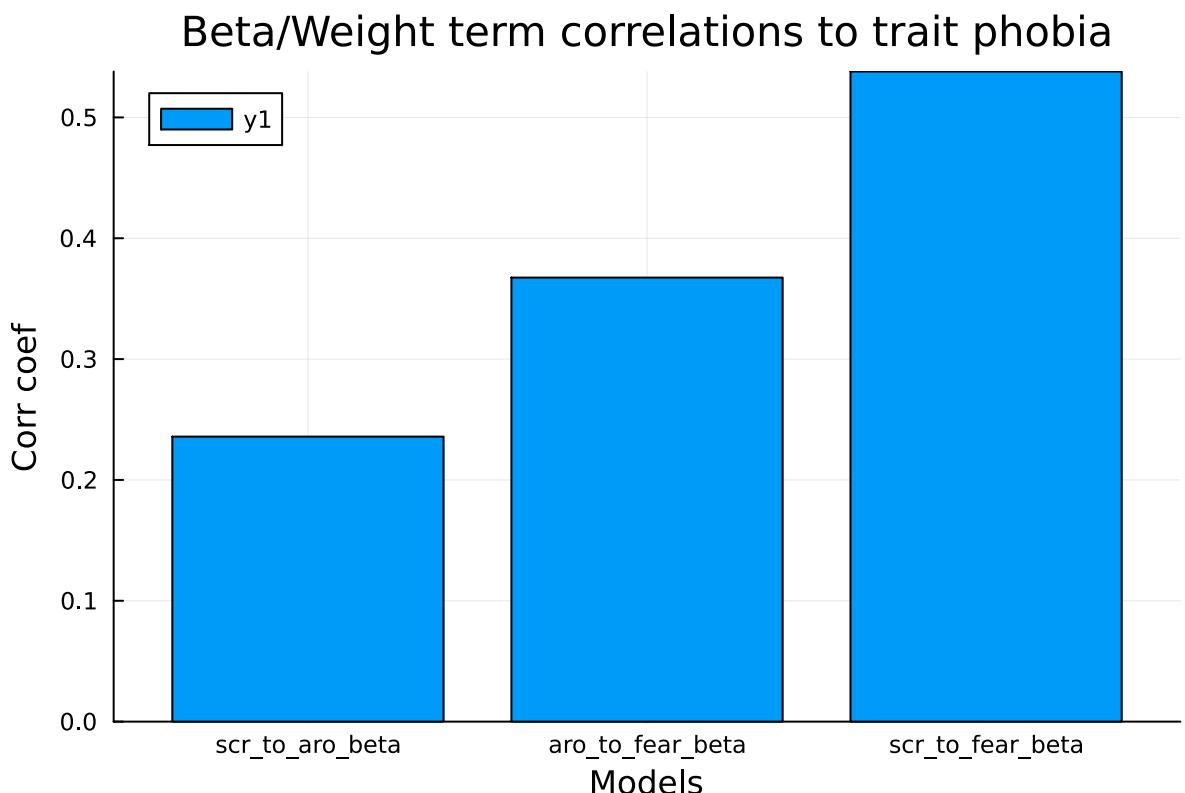
```
1 plotcols3 = ["scr_to_aro_int","aro_to_fear_int", "scr_to_fear_int"];
```



```
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];
```

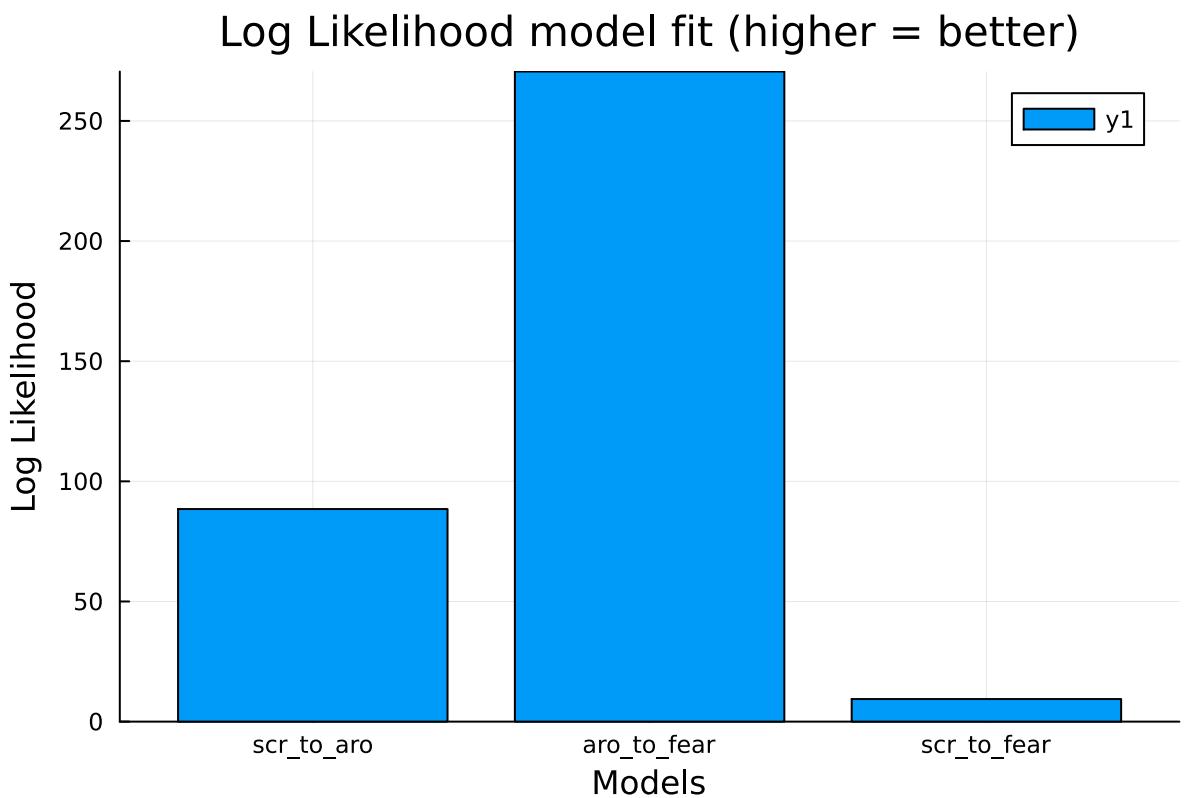
```
1 plotcols4 = ["scr_to_aro_beta","aro_to_fear_beta", "scr_to_fear_beta"];
```



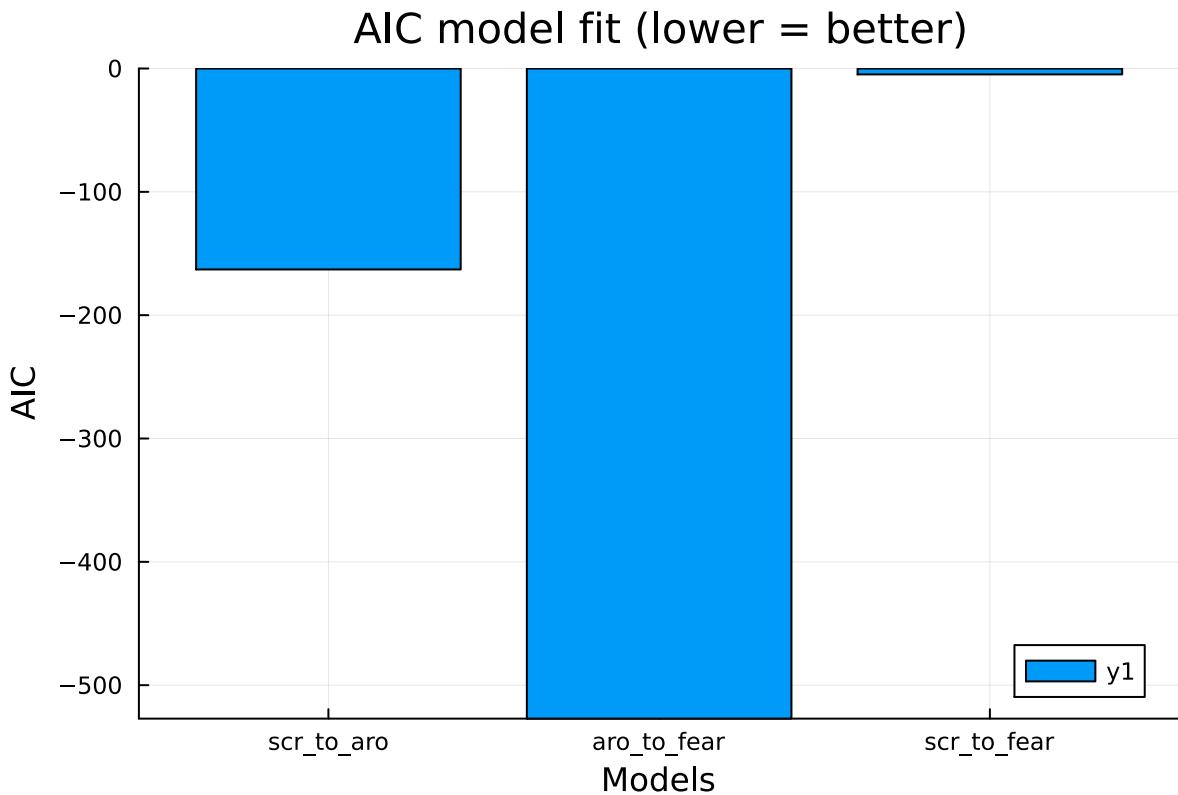
```
1 plot(bar(plotcols4, plotdata4), title="Beta/Weight term correlations to trait phobia", xlabel="Models", ylabel="Corr coef")
```

```
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 =
```

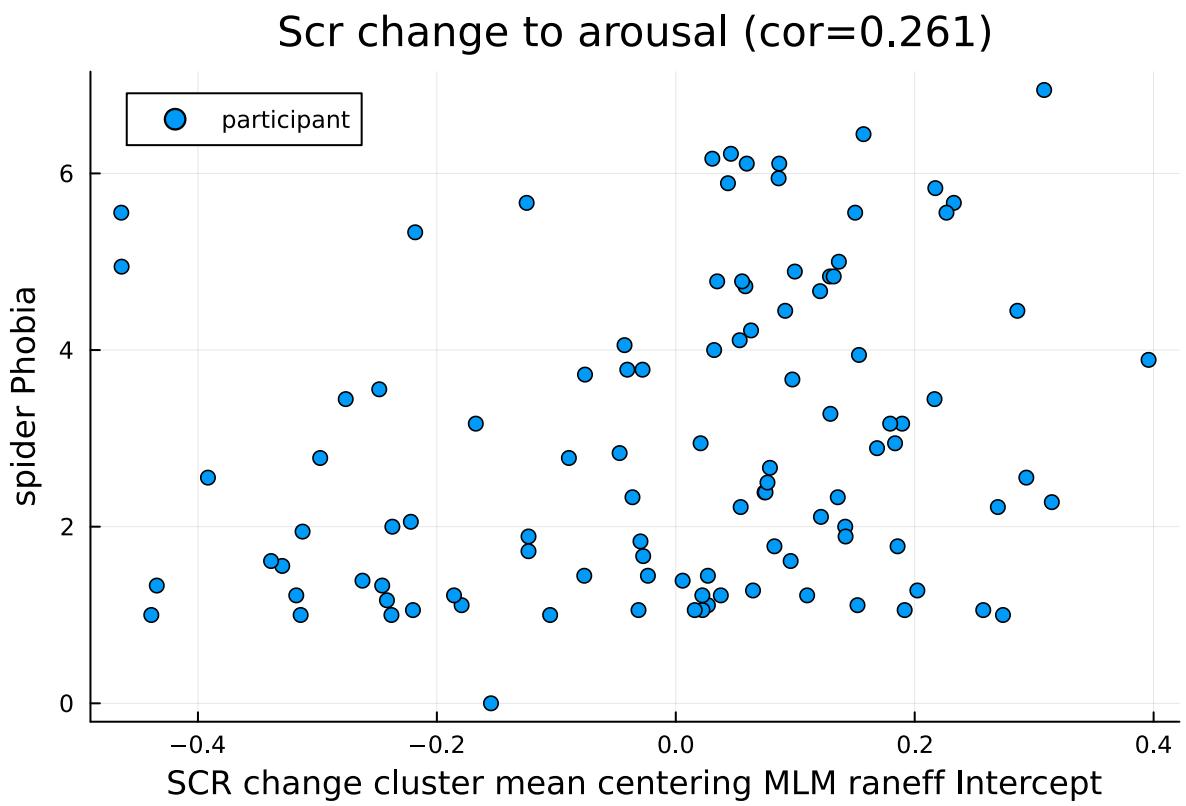
	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.4929	0.0211	23.37	<1e-99	0.2014
scr_change_video_cmc	0.3625	0.1585	2.29	0.0222	0.4802
scr_change_video_mean	0.4528	0.6904	0.66	0.5119	
Residual	0.2024				

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

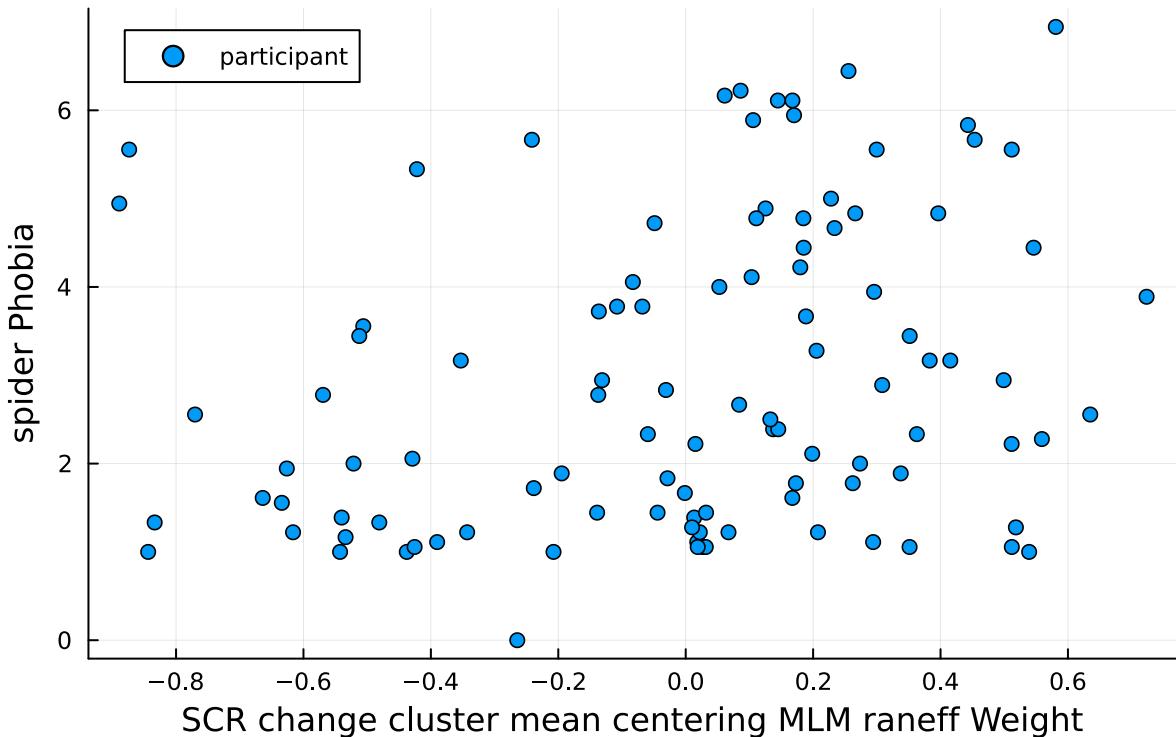
```
85.43209208594354
```

```
1 loglikelihood(mm5)
```

-156.86418417188708



## Scr change to arousal (cor=0.262)



## Scr change to Fear

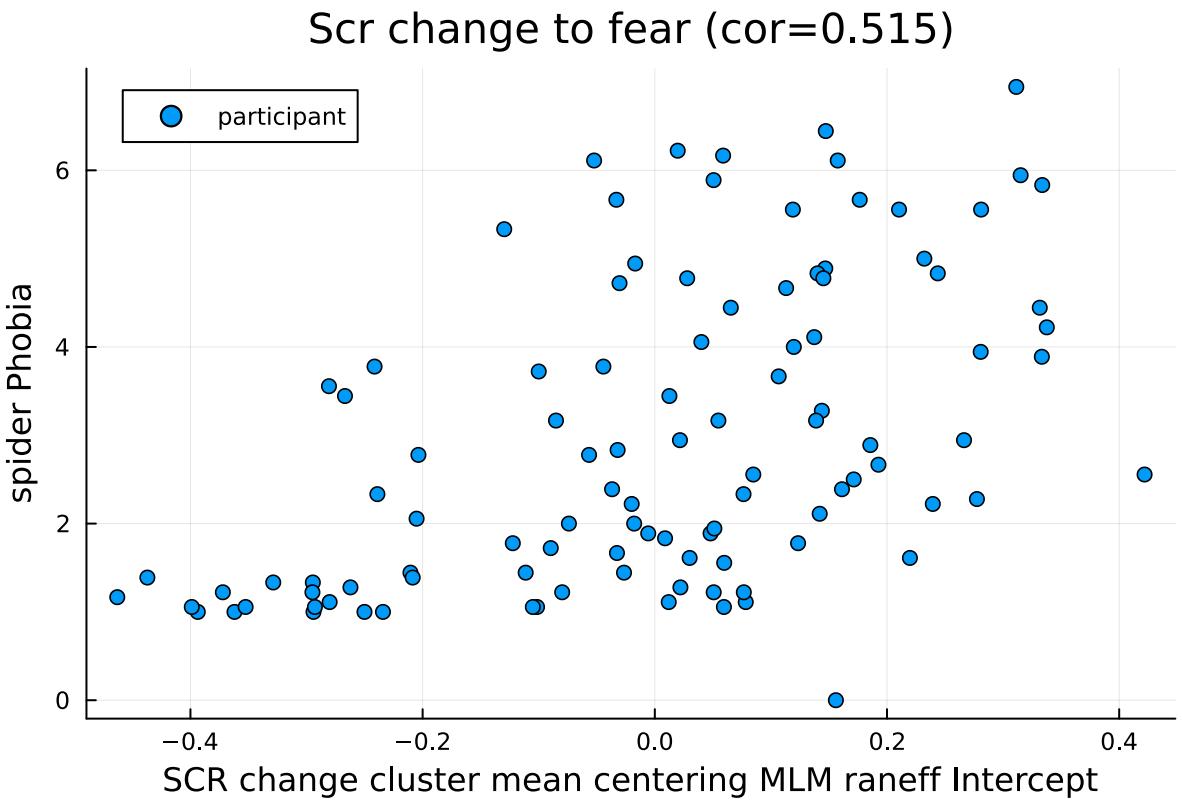
```
mm7 =
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.4514	0.0221	20.41	<1e-91	0.2106
scr_change_video_cmc	0.5188	0.1616	3.21	0.0013	0.2494
scr_change_video_mean	1.6034	0.7373	2.17	0.0296	
Residual	0.2161				

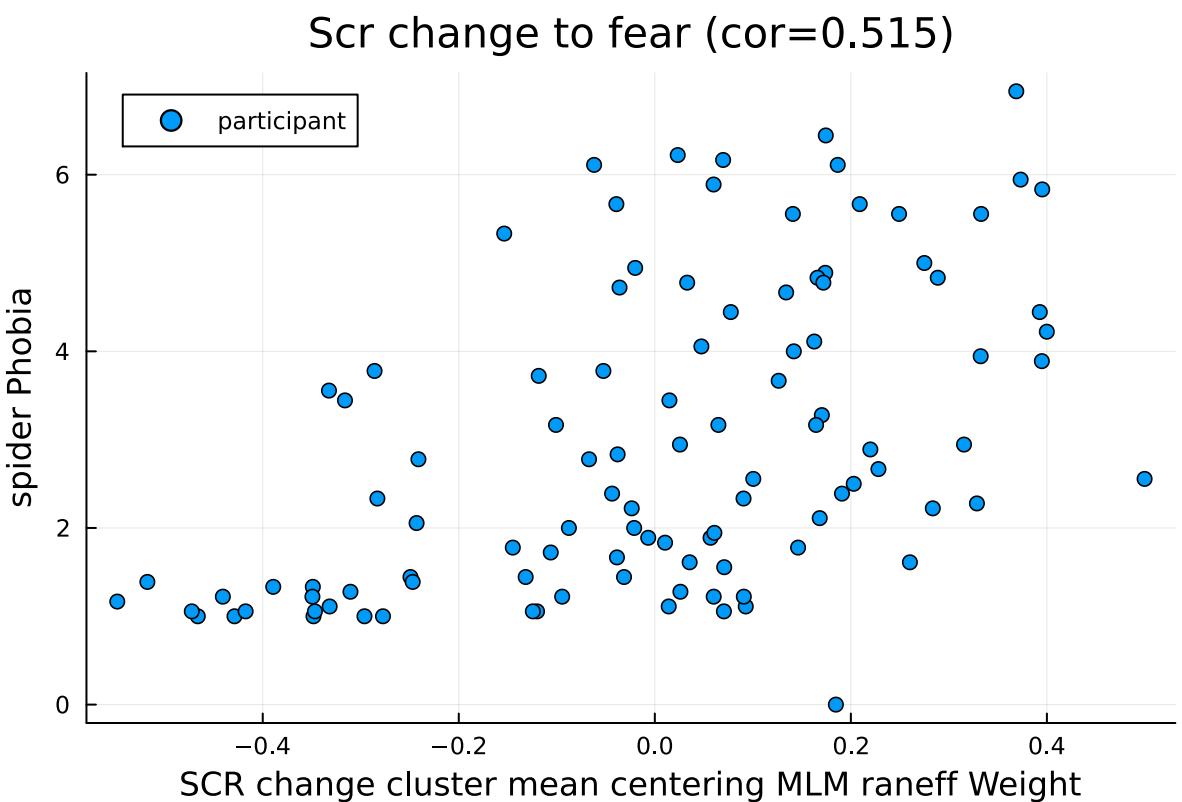
```
1 mm7 = fit(LinearMixedModel, @formula(resp_fear ~ scr_change_video_cmc +
scr_change_video_mean + (scr_change_video_cmc|participant_num_str)), dfcleancol2)
```

8.289584609172834

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



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



## Scr to Arousal with spider phobia modulating term

```
mm8 =
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.3442	0.0436	7.89	<1e-14	0.1876
video_scr_cmc	0.1598	0.3080	0.52	0.6039	0.4368
spider_phobia	0.0285	0.0115	2.48	0.0132	
video_scr_mean	1.8766	0.8417	2.23	0.0258	
video_scr_cmc & spider_phobia	0.0842	0.0904	0.93	0.3519	
Residual	0.2024				

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

```
91.58742203030903
```

```
1 loglikelihood(mm8)
```

```
-165.17484406061806
```

```
1 aic(mm8)
```

```
mm8coef_i = 7.410857933628743e-16
```

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

```
mm8coef_w = 7.75088773405599e-16
```

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

```
mm9 =
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.0964	0.0444	2.17	0.0300	0.1599
resp_arousal_cmc	0.3490	0.0931	3.75	0.0002	0.3720
spider_phobia	0.0601	0.0099	6.06	<1e-08	
resp_arousal_mean	0.3399	0.0774	4.39	<1e-04	
resp_arousal_cmc & spider_phobia	0.0805	0.0280	2.87	0.0041	
Residual	0.1659				

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

```
287.31295310294246
```

```
1 loglikelihood(mm9)
```

```
mm10 =
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.1840	0.0410	4.49	<1e-05	0.1731
video_scr_cmc	-0.0263	0.3114	-0.08	0.9326	0.1037
spider_phobia	0.0665	0.0108	6.17	<1e-09	
video_scr_mean	1.8327	0.7956	2.30	0.0213	
video_scr_cmc & spider_phobia	0.1898	0.0913	2.08	0.0375	
Residual	0.2159				

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

```
27.895105782370962
```

```
1 loglikelihood(mm10)
```

```
mm11 =
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.0850	0.0452	1.88	0.0601	0.1589
video_scr_cmc	0.0166	0.2493	0.07	0.9468	0.1527
spider_phobia	0.0580	0.0099	5.84	<1e-08	
resp_arousal_cmc	0.3326	0.0931	3.57	0.0004	0.3709
video_scr_mean	0.8246	0.7022	1.17	0.2403	
resp_arousal_mean	0.3182	0.0790	4.03	<1e-04	
video_scr_cmc & spider_phobia	0.0646	0.0739	0.87	0.3820	
video_scr_cmc & resp_arousal_cmc	-0.5926	1.4815	-0.40	0.6892	2.2332
spider_phobia & resp_arousal_cmc	0.0840	0.0281	2.99	0.0028	
video_scr_cmc & spider_phobia & resp_arousal_cmc	0.4735	0.4639	1.02	0.3073	
Residual	0.1648				

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

```
Minimizing 1229    Time: 0:00:00 (81.37 µs/it)
objective: -588.6168856868518
```

```
Minimizing 2153    Time: 0:00:00 (72.92 µs/it)
```

```
Minimizing 2154    Time: 0:00:00 (72.89 µs/it)
objective: -588.6258636267605
```

mm13 =

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.3282	0.0500	6.56	<1e-10	0.1268
spider_phobia	0.0425	0.0082	5.18	<1e-06	
video_scr_cmc	-0.3501	0.2204	-1.59	0.1122	0.3109
resp_arousal_cmc	0.3606	0.0656	5.50	<1e-07	0.2050
resp_valence_cmc	-0.4973	0.0662	-7.51	<1e-13	0.2328
video_scr_mean	-0.1461	0.5351	-0.27	0.7848	
resp_arousal_mean	0.3255	0.0591	5.51	<1e-07	
resp_valence_mean	-0.5721	0.0769	-7.44	<1e-12	
spider_phobia & video_scr_cmc	0.1149	0.0665	1.73	0.0840	
spider_phobia & resp_arousal_cmc	0.0302	0.0202	1.50	0.1338	
video_scr_cmc & resp_arousal_cmc	-1.2231	1.2573	-0.97	0.3307	1.0564
spider_phobia & resp_valence_cmc	-0.0150	0.0212	-0.71	0.4790	
video_scr_cmc & resp_valence_cmc	-2.2798	1.2801	-1.78	0.0749	0.2396
resp_arousal_cmc & resp_valence_cmc	-0.5415	0.2144	-2.53	0.0116	0.3249
spider_phobia & video_scr_cmc & resp_arousal_cmc	0.4853	0.4165	1.17	0.2440	
spider_phobia & video_scr_cmc & resp_valence_cmc	0.4775	0.4739	1.01	0.3137	
spider_phobia & resp_arousal_cmc & resp_valence_cmc	0.0420	0.0669	0.63	0.5299	
video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	6.7272	6.0275	1.12	0.2644	7.3919
spider_phobia & video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	-0.9320	2.0994	-0.44	0.6571	
Residual	0.1343				

```
1 mm13= fit(LinearMixedModel, @formula(resp_fear ~ spider_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 592      Time: 0:00:00 ( 0.17 ms/it )



```
objective: -1070.870971720845

Minimizing 1308    Time: 0:00:00 ( 0.15 ms/it)
objective: -1072.8036816896501

Minimizing 2024    Time: 0:00:00 ( 0.15 ms/it)
objective: -1073.3260906850194

Minimizing 2740    Time: 0:00:00 ( 0.15 ms/it)
objective: -1074.3503319324755

Minimizing 3447    Time: 0:00:00 ( 0.15 ms/it)
objective: -1075.2606673685546

Minimizing 4149    Time: 0:00:00 ( 0.15 ms/it)
objective: -1075.9767142993164

Minimizing 4857    Time: 0:00:00 ( 0.15 ms/it)
objective: -1076.3286581694533
```

294.31293181338026

```
1 loglikelihood(mm11)
```

538.3228492158562

```
1 loglikelihood(mm13)
```

-546.6258636267605

```
1 aic(mm11)
```

-964.6456984317124

```
1 aic(mm13)
```

mm14 =

	<b>Est.</b>	<b>SE</b>	<b>z</b>	<b>p</b>	<b>σ_participant_num_str</b>
(Intercept)	0.3448	0.0545	6.33	<1e-09	0.1252
spider_phobia	0.0416	0.0081	5.11	<1e-06	
video_scr_cmc	-0.3329	0.2243	-1.48	0.1378	0.3505
video_hp_cmc	0.0377	0.0481	0.78	0.4337	0.0474
resp_arousal_cmc	0.3904	0.0640	6.10	<1e-08	0.1927
resp_valence_cmc	-0.5035	0.0669	-7.53	<1e-13	0.2367
video_scr_mean	-0.0854	0.5161	-0.17	0.8686	
video_hp_mean	-0.0452	0.0409	-1.10	0.2701	
resp_arousal_mean	0.3462	0.0564	6.14	<1e-09	
resp_valence_mean	-0.5487	0.0738	-7.43	<1e-12	
spider_phobia & video_scr_cmc	0.1191	0.0675	1.76	0.0777	
spider_phobia & video_hp_cmc	-0.0090	0.0144	-0.62	0.5340	
video_scr_cmc & video_hp_cmc	1.7057	1.4955	1.14	0.2540	1.0491
spider_phobia & resp_arousal_cmc	0.0244	0.0197	1.24	0.2160	
video_scr_cmc & resp_arousal_cmc	-1.0357	1.2480	-0.83	0.4066	0.4742
video_hp_cmc & resp_arousal_cmc	-0.0460	0.3951	-0.12	0.9073	0.3612
spider_phobia & resp_valence_cmc	-0.0109	0.0214	-0.51	0.6121	
video_scr_cmc & resp_valence_cmc	-2.1304	1.2756	-1.67	0.0949	0.2053
video_hp_cmc & resp_valence_cmc	-0.0543	0.3254	-0.17	0.8674	0.3070
resp_arousal_cmc & resp_valence_cmc	-0.5932	0.2157	-2.75	0.0060	0.3310
spider_phobia & video_scr_cmc & video_hp_cmc	-0.5565	0.4551	-1.22	0.2214	
spider_phobia & video_scr_cmc & resp_arousal_cmc	0.4553	0.4125	1.10	0.2697	
spider_phobia & video_hp_cmc & resp_arousal_cmc	-0.0212	0.1261	-0.17	0.8662	
video_scr_cmc & video_hp_cmc & resp_arousal_cmc	7.7648	8.1339	0.95	0.3398	2.5310
spider_phobia & video_scr_cmc & resp_valence_cmc	0.5081	0.4703	1.08	0.2799	

spider_phobia & video_hp_cmc & resp_valence_cmc	-0.0494	0.1106	-0.45	0.6550	
video_scr_cmc & video_hp_cmc & resp_valence_cmc	4.8434	8.7620	0.55	0.5804	8.8061
spider_phobia & resp_arousal_cmc & resp_valence_cmc	0.0507	0.0683	0.74	0.4583	
video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	5.1013	6.2544	0.82	0.4147	10.7330
video_hp_cmc & resp_arousal_cmc & resp_valence_cmc	1.7681	1.7996	0.98	0.3258	0.9641
spider_phobia & video_scr_cmc & video_hp_cmc & resp_arousal_cmc	-0.7022	2.7066	-0.26	0.7953	
spider_phobia & video_scr_cmc & video_hp_cmc & resp_valence_cmc	-3.1121	3.0953	-1.01	0.3147	
spider_phobia & video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	-0.5170	2.1943	-0.24	0.8137	
spider_phobia & video_hp_cmc & resp_arousal_cmc & resp_valence_cmc	-0.3081	0.6773	-0.45	0.6492	
video_scr_cmc & video_hp_cmc & resp_arousal_cmc & resp_valence_cmc	10.9594	38.0877	0.29	0.7735	3.9001
spider_phobia & video_scr_cmc & video_hp_cmc & resp_arousal_cmc & resp_valence_cmc	-6.0857	13.8640	-0.44	0.6607	
Residual	0.1318				

```
1 mm14 = fit(LinearMixedModel, @formula(resp_fear ~ spider_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 249 Time: 0:00:00 ( 0.41 ms/it)  
 objective: -1014.1192286067198

```
Minimizing 492      Time: 0:00:00 ( 0.61 ms/it)
objective: -1082.5491792602156
```

```
Minimizing 615      Time: 0:00:00 ( 0.67 ms/it)
objective: -1084.5858075333729
```

```
Minimizing 726      Time: 0:00:00 ( 0.70 ms/it)
objective: -1086.4809942572056
```

```
Minimizing 837      Time: 0:00:00 ( 0.73 ms/it)
objective: -1088.012976473629
```

```
Minimizing 945      Time: 0:00:00 ( 0.76 ms/it)
objective: -1088.931533226702
```

```
Minimizing 1051     Time: 0:00:00 ( 0.78 ms/it)
objective: -1089.6263034579454
```

```
mm15 =
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.3536	0.0573	6.17	<1e-09	0.1262
spider_phobia	0.0415	0.0082	5.07	<1e-06	
video_scr_cmc	-0.3036	0.2074	-1.46	0.1432	0.1554
video_hp_cmc	0.0228	0.0455	0.50	0.6168	0.0576
resp_arousal_cmc	0.3255	0.0616	5.28	<1e-06	0.1867
resp_valence_cmc	-0.5283	0.0691	-7.64	<1e-13	0.2450
video_scr_mean	0.0330	0.5403	0.06	0.9512	
video_hp_mean	-0.0721	0.0443	-1.63	0.1034	
resp_arousal_mean	0.3487	0.0588	5.93	<1e-08	
resp_valence_mean	-0.5209	0.0766	-6.80	<1e-10	
spider_phobia & video_scr_cmc	0.1151	0.0607	1.90	0.0580	
spider_phobia & video_hp_cmc	-0.0045	0.0136	-0.33	0.7386	
spider_phobia & resp_arousal_cmc	0.0370	0.0192	1.93	0.0538	
spider_phobia & resp_valence_cmc	-0.0043	0.0220	-0.20	0.8454	
Residual	0.1366				

```
1 mm15 = fit(LinearMixedModel, @formula(resp_fear ~ spider_phobia * video_scr_cmc + spider_phobia * video_hp_cmc + spider_phobia * resp_arousal_cmc + spider_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 892    Time: 0:00:00 ( 0.11 ms/it)
objective: -1044.3578240140623
```

```
Minimizing 1334    Time: 0:00:00 ( 0.10 ms/it)
```

```
Minimizing 1335    Time: 0:00:00 ( 0.10 ms/it)
objective: -1044.3607328218104
```

mm16 =

	Est.	SE	z	p	σ_participant_num_str
(Intercept)	0.3645	0.0531	6.86	<1e-11	0.1262
spider_phobia	0.0392	0.0077	5.11	<1e-06	
video_scr_cmc	-0.4150	0.2147	-1.93	0.0532	0.3367
video_hp_cmc	0.0123	0.0442	0.28	0.7816	0.0446
resp_arousal_cmc	0.3257	0.0594	5.48	<1e-07	0.2094
resp_valence_cmc	-0.5179	0.0654	-7.92	<1e-14	0.2295
video_scr_mean	-0.1339	0.5074	-0.26	0.7918	
video_hp_mean	-0.0526	0.0401	-1.31	0.1890	
resp_arousal_mean	0.3781	0.0567	6.67	<1e-10	
resp_valence_mean	-0.5652	0.0739	-7.65	<1e-13	
spider_phobia & video_scr_cmc	0.1316	0.0625	2.10	0.0354	
spider_phobia & video_hp_cmc	-0.0034	0.0133	-0.25	0.8008	
spider_phobia & resp_arousal_cmc	0.0328	0.0187	1.75	0.0800	
spider_phobia & resp_valence_cmc	-0.0077	0.0209	-0.37	0.7129	
resp_arousal_cmc & resp_valence_cmc					0.5231
video_hp_cmc & resp_arousal_cmc & resp_valence_cmc					0.8047
video_scr_cmc & video_hp_cmc & resp_valence_cmc					7.3747
video_scr_cmc & video_hp_cmc					1.0638
video_scr_cmc & video_hp_cmc & resp_arousal_cmc					6.3066
video_scr_cmc & resp_valence_cmc					1.0392
video_scr_cmc & resp_arousal_cmc					1.1877
video_hp_cmc & resp_arousal_cmc					0.3587
video_scr_cmc & resp_arousal_cmc & resp_valence_cmc					11.9929
video_scr_cmc & video_hp_cmc & resp_arousal_cmc & resp_valence_cmc					3.9770
video_hp_cmc &					0.2541

```
resp_valence_cmc  
Residual          0.1323
```

```
1 mm16 = fit(LinearMixedModel, @formula(resp_fear ~ spider_phobia * video_scr_cmc  
+ spider_phobia * video_hp_cmc + spider_phobia * resp_arousal_cmc +  
spider_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 ( 0.44 ms/it)  
objective: -1051.0125185499003

Minimizing 664      Time: 0:00:00 ( 0.57 ms/it)  
objective: -1057.7748442662887

Minimizing 787      Time: 0:00:00 ( 0.61 ms/it)  
objective: -1059.2078121704187

Minimizing 908      Time: 0:00:00 ( 0.66 ms/it)  
objective: -1060.2553098238288

Minimizing 1026     Time: 0:00:00 ( 0.68 ms/it)  
objective: -1061.3870581395952

Minimizing 1145     Time: 0:00:00 ( 0.69 ms/it)  
objective: -1062.1511029900253

Minimizing 1261     Time: 0:00:00 ( 0.72 ms/it)  
objective: -1063.0973371429313
```

## Plot Test

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_a
1	"spider_high_5.m4v"	107	0.248	3.22	0.125
2	"spider_low_3.m4v"	107	0.318	2.636	0.141
3	"spider_low_6.m4v"	107	0.347	3.571	0.227
4	"spider_high_6.mov"	107	0.225	2.086	0.126
5	"spider_low_4.m4v"	107	0.052	1.485	0.288
6	"spider_high_1.mov"	107	0.267	2.57	0.218
7	"spider_high_4.m4v"	107	0.311	5.623	0.269
8	"spider_low_1.m4v"	107	0.306	7.525	0.31
9	"spider_low_2.mov"	107	0.044	1.435	0.06
10	"spider_high_3.mov"	107	0.141	1.902	0.102
: more					
1234	"spider_low_5.mov"	232	0.165	32.386	0.091

```
1 dfcleancol2
```

► [0.787, 0.872, 0.76, 0.772, 0.699, 0.747, 0.951, 0.5, 0.707, 0.907, 0.923, 0.059, 0.858]

```
1 dfcleancol2.resp_arousal
```

	Est.	SE	z	p	$\sigma_{\text{participant\_num\_str}}$
(Intercept)	0.4153	0.0341	12.19	<1e-33	0.1932
video_scr_cmc	0.4128	0.1586	2.60	0.0093	0.4652
video_scr_mean	2.2638	0.8460	2.68	0.0075	
Residual	0.2024				

```
1 mm3
```

```
randeff =
2×105 Matrix{Float64}:
0.182984 0.131778 0.144233 -0.0374075 0.194484 ...
0.486004 0.154168 0.08586 0.0166045 0.281367 ...
0.240613 0.26485 -0.0243491
0.376944 0.44336 -0.0234263
```

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

```
► [0.182984, 0.131778, 0.144233, -0.0374075, 0.194484, 0.191161, 0.0378006, 0.030787, -0
```

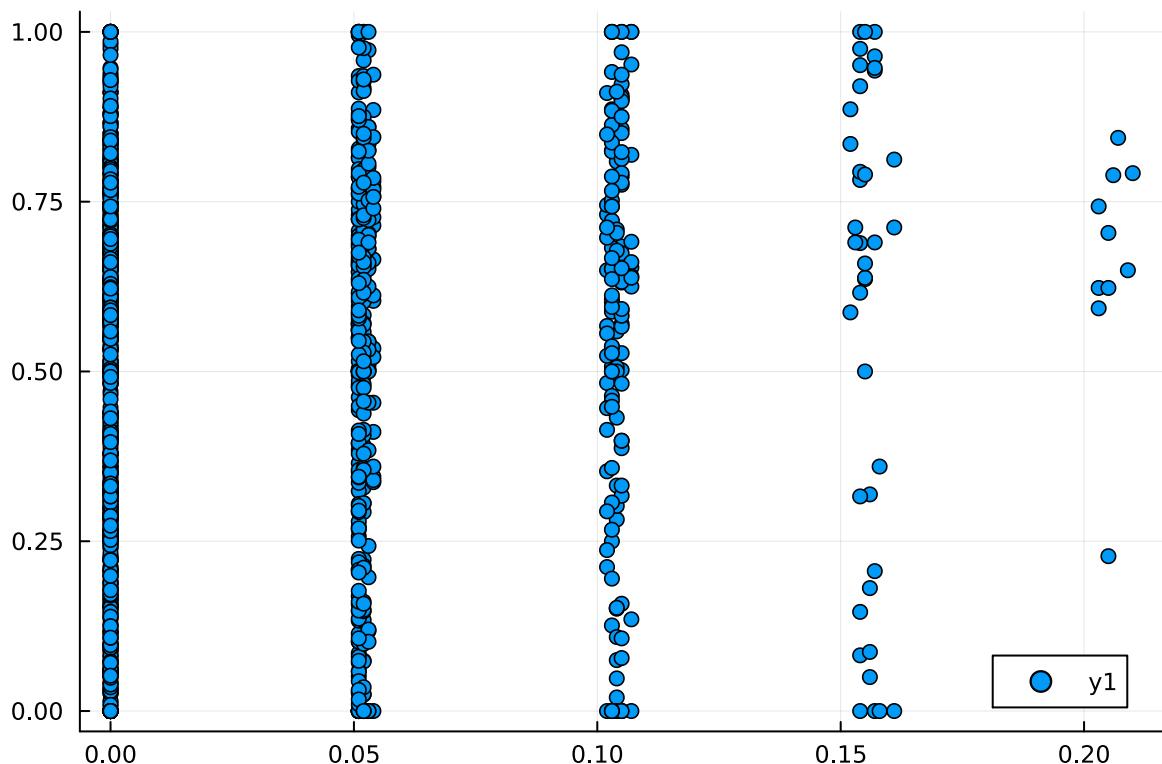
```
1 randeff[1,:]
```

```
2×1 Matrix{Float64}:
```

```
0.18562230130908697
```

```
0.30596341072770716
```

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



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

```
0.21
```

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

```
0.0
```

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

```
xax = 10-element LinRange{Float64, Int64}:
```

```
0.0, 0.0233333, 0.0466667, 0.07, 0.0933333, 0.116667, 0.14, 0.163333, 0.186667, 0.21
```

```
1 xax = LinRange(minimum(dfcleancol2.video_scr),maximum(dfcleancol2.video_scr),10)
```

```
inter = 0.41526869960952023
```

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

```
cmcbeta = 0.4127630120937151
```

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

```
meanbeta = 2.2637635991925205
```

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

```
yax =  
10-element LinRange{Float64, Int64}:  
0.415269, 0.477721, 0.540173, 0.602626, 0.665078, ..., 0.789982, 0.852435, 0.914887, 0.977339  
1 yax = inter .+ ((cmcbeta .* xax) + (meanbeta .* xax))
```

```
► [1.27778, 5.0, 1.77778, 1.66667, 2.88889, 4.66667, 2.38889, 1.11111, 1.44444, 1.0, 2.5
```

```
1 df2_subjlvl_hp.spider_phobia
```

```
1.7146404133042563
```

```
1 std(df2_subjlvl_hp.spider_phobia)
```

```
std_up = 4.658026656690501
```

```
1 std_up = mean(df2_subjlvl_hp.spider_phobia) + std(df2_subjlvl_hp.spider_phobia)
```

```
bm_std_up =
```

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

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

```
0.0340003105961937
```

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

```
std_dwn = 1.2287458300819876
```

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

```
bm_std_dwn =
```

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

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

```
-0.06001034693909727
```

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

```
yax_up =
```

```
10-element LinRange{Float64, Int64}:  
0.0340003, 0.0355208, 0.0370414, 0.0385619, ..., 0.0431235, 0.0446441, 0.0461646, 0.0476852
```

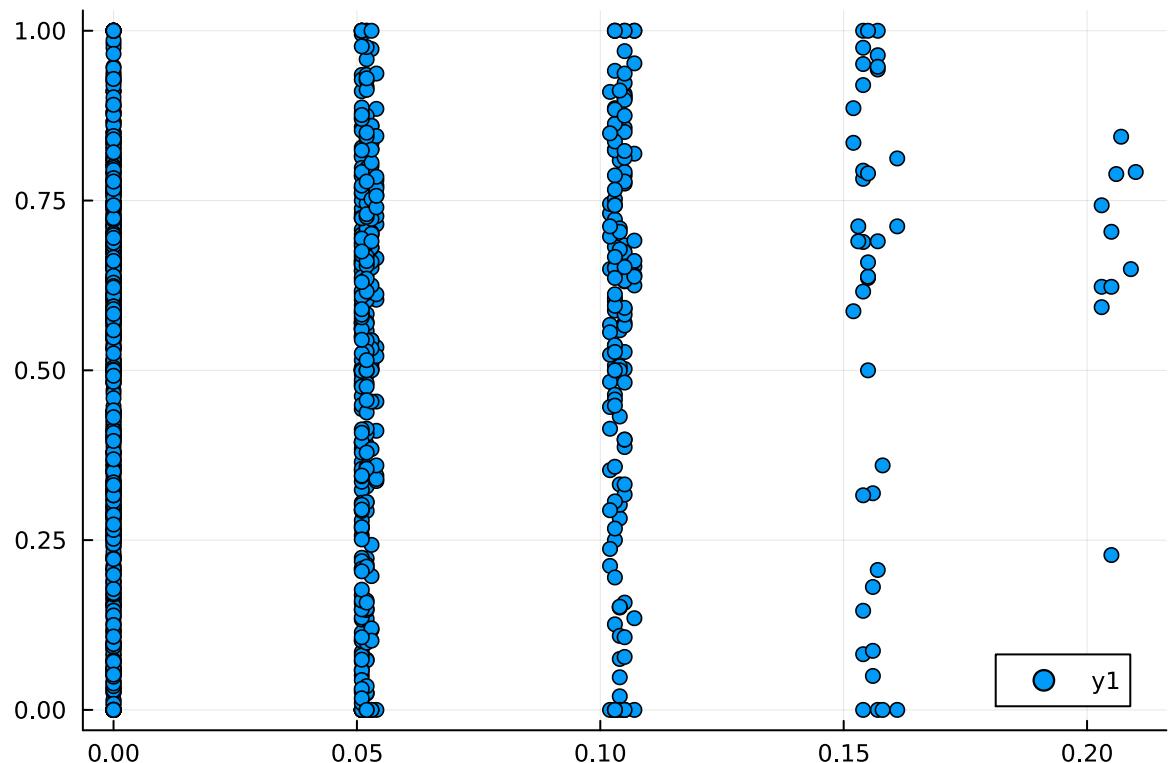
```
1 yax_up = mean(randeff[1,bm_std_up]) .+ (mean(randeff[2,bm_std_up]).* xax)
```

```
yax_dwn =  
► [0.0600103, 0.062208, 0.0644056, 0.0666032, 0.0688008, 0.0709984, 0.073196, 0.0753937,  
1 yax_dwn = abs.(mean(randeff[1,bm_std_dwn]) .+ (mean(randeff[2,bm_std_dwn]).*  
xax))
```

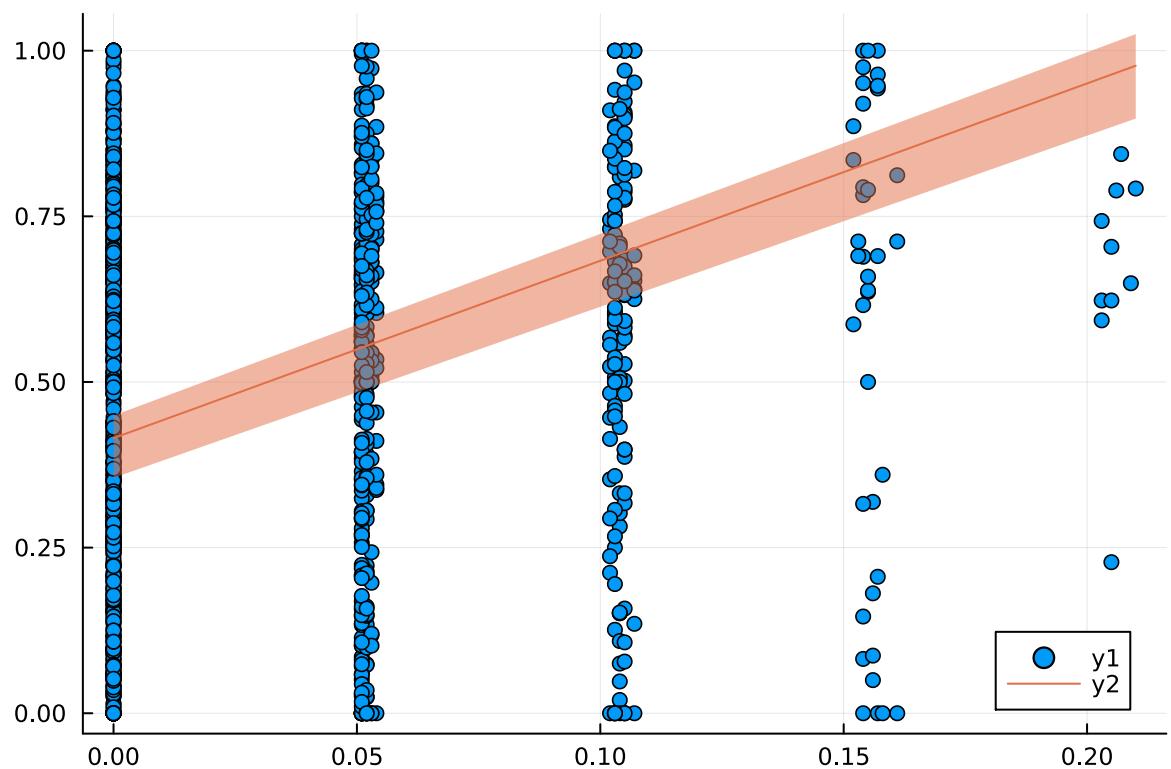
df =

	xax	yax	yax_up	yax_dwn
1	0.0	0.415269	0.0340003	0.0600103
2	0.0233333	0.477721	0.0355208	0.062208
3	0.0466667	0.540173	0.0370414	0.0644056
4	0.07	0.602626	0.0385619	0.0666032
5	0.0933333	0.665078	0.0400825	0.0688008
6	0.1166667	0.72753	0.041603	0.0709984
7	0.14	0.789982	0.0431235	0.073196
8	0.163333	0.852435	0.0446441	0.0753937
9	0.1866667	0.914887	0.0461646	0.0775913
10	0.21	0.977339	0.0476852	0.0797889

```
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))
```

	<b>Est.</b>	<b>SE</b>	<b>z</b>	<b>p</b>	<b>σ_participant_num_str</b>
(Intercept)	0.4153	0.0341	12.19	<1e-33	0.1932
video_scr_cmc	0.4128	0.1586	2.60	0.0093	0.4652
video_scr_mean	2.2638	0.8460	2.68	0.0075	
Residual	0.2024				

1 mm3

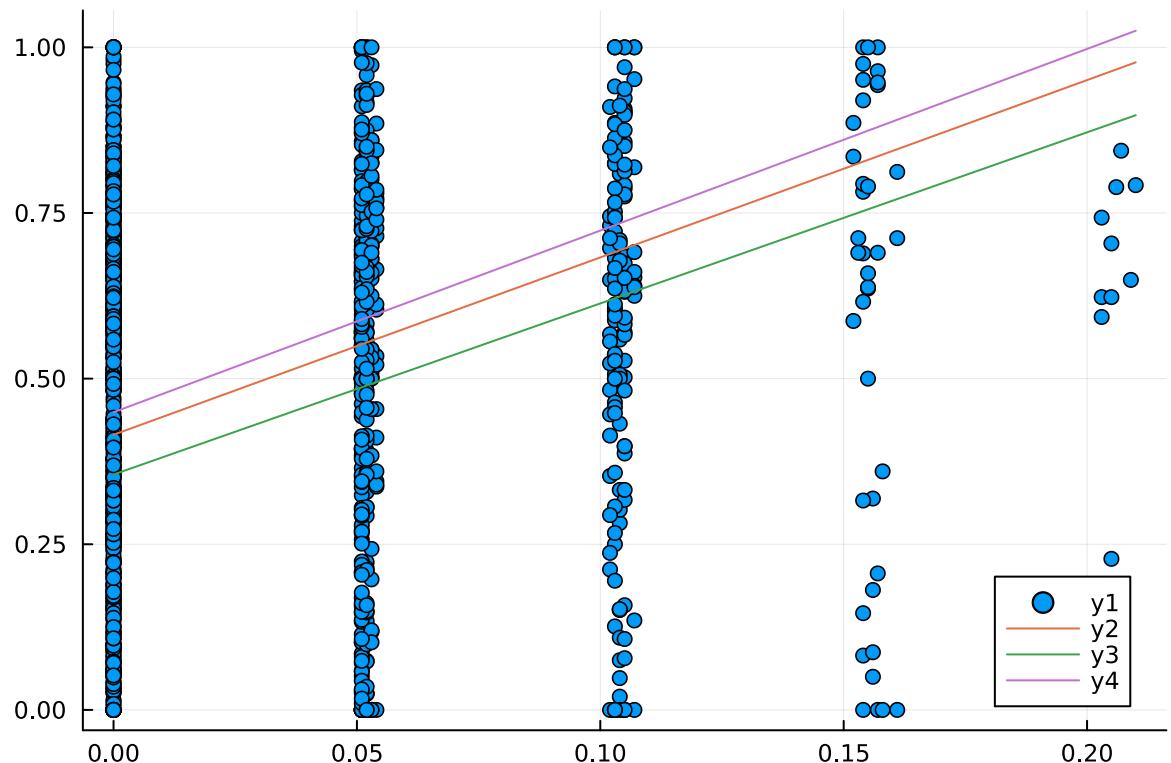
▶ [2×105 Matrix{Float64}]:  
0.182984 0.131778 0.144233 -0.0374075 0.194484 ... 0.240613 0.26485 -0.024345  
1 ranef(mm3)

1 Enter cell code...

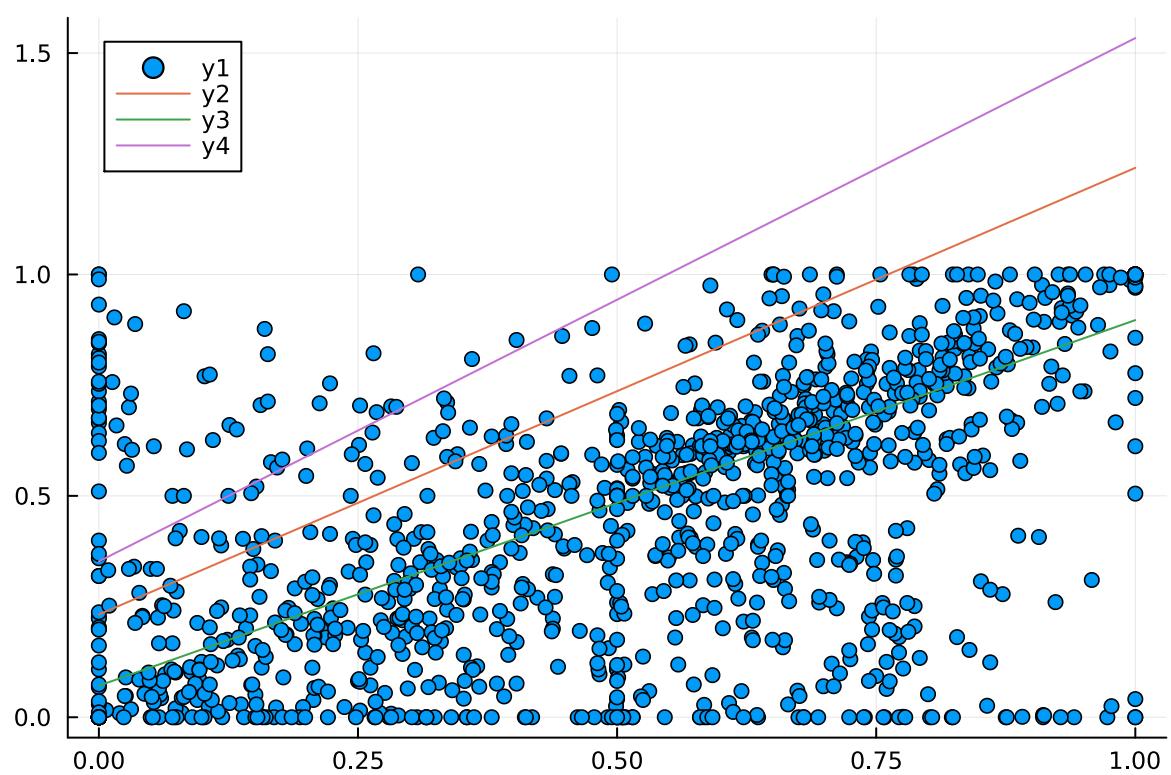
## Plot Generic

plot\_mm (generic function with 1 method)

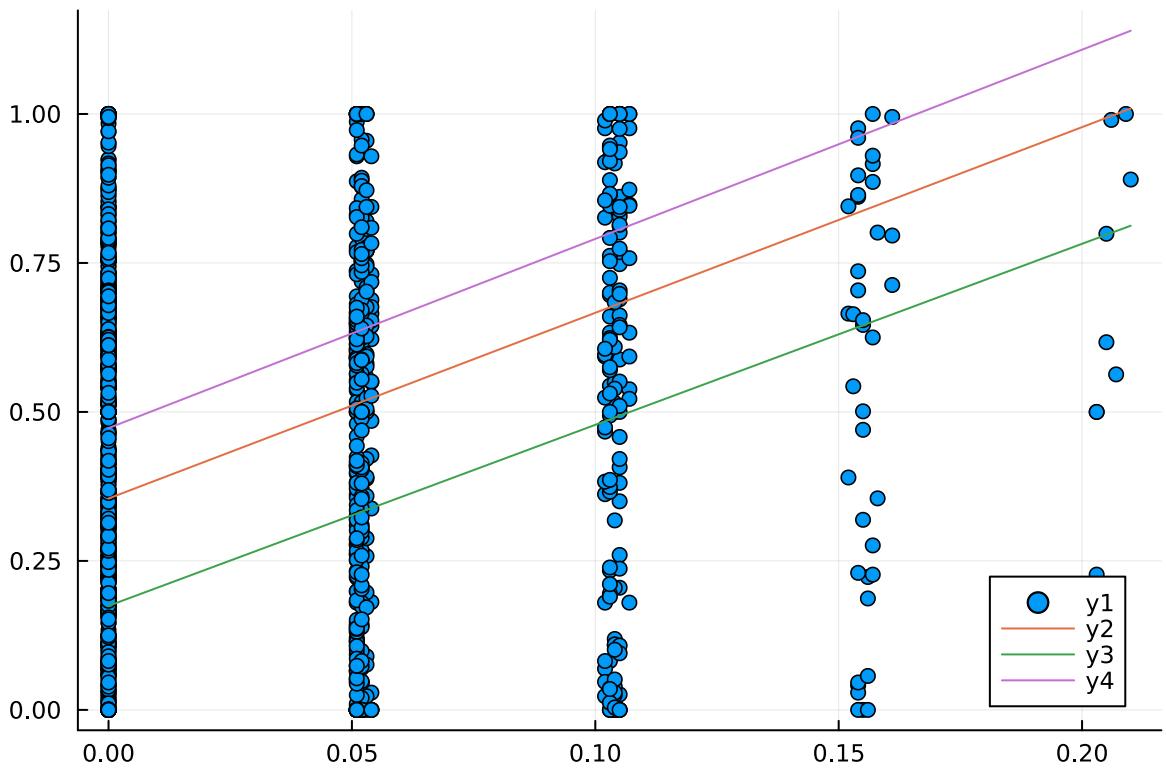
```
1 function plot_mm(mm, varx, vary)
2     randeff = only(ranef(mm))
3     xax = LinRange(minimum(varx),maximum(varx),10)
4     inter = coef(mm)[1]
5     cmcbeta = coef(mm)[2]
6     meanbeta = coef(mm)[3]
7     yax = inter .+ ((cmcbeta .* xax) + (meanbeta .* xax))
8     std_up = mean(df2_subjlvl_hp.spider_phobia) +
9     std(df2_subjlvl_hp.spider_phobia)
10    bm_std_up = df2_subjlvl_hp.spider_phobia .> std_up
11    std_dwn = mean(df2_subjlvl_hp.spider_phobia) -
12    std(df2_subjlvl_hp.spider_phobia)
13    bm_std_dwn = df2_subjlvl_hp.spider_phobia .< std_dwn
14    yax_up = yax + (mean(randeff[1,bm_std_up]) .+ (mean(randeff[2,bm_std_up]).*
15    xax))
16    yax_dwn = yax + (mean(randeff[1,bm_std_dwn]) .+
17    (mean(randeff[2,bm_std_dwn]).* xax))
18    df = DataFrame(xax = xax, yax = yax, yax_up = yax_up, yax_dwn = yax_dwn)
19    scatter(varx, vary)
20    @df df plot!(:xax, :yax)
21    @df df plot!(:xax, :yax_dwn)
22    @df df plot!(:xax, :yax_up)
23 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, :spider_phobia => mean, renamecols=false).spider_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="spider 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="spider 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="spider 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="spider 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(mgz3))"
```

```
UndefVarError: mmz3 not defined
```

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

```
1 mmz3coef_w = cor(ranef(mgz3)[1][2,:], hparr)
```

```
UndefVarError: mmz3 not defined
```

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

```
1 mmz3coef_i = cor(ranef(mgz3)[1][1,:], hparr)
```

```
UndefVarError: mmz3 not defined
```

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

```
1 scatter(ranef(mgz3)[1][1,:], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(mgz3coef_i)))", xlabel="SCR cluster mean centering MLM raneff  
Intercept", ylabel="spider Phobia")
```

```
UndefVarError: mmz3 not defined
```

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

```
1 scatter(ranef(mgz3)[1][2,:], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(mgz3coef_w)))", xlabel="SCR cluster mean centering MLM raneff  
Weight", ylabel="spider 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")
```