

Modeling Notebook

1 md"# Modeling Notebook"

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

cond_num = 1

```
1 cond_num = 1
```

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

```
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: o	video	video_condition	im_condition	pred_
1	3	3	"heights_low_3.mov"	1	1	2
2	8	8	"heights_low_2.m4v"	1	1	2
3	9	9	"heights_high_1.m4v"	1	2	1
4	10	10	"heights_high_3.m4v"	1	2	1
5	14	14	"heights_high_5.mov"	1	2	2
6	15	15	"heights_high_4.m4v"	1	2	1
7	16	16	"heights_low_6.m4v"	1	1	1
8	20	20	"heights_low_1.m4v"	1	1	2
9	25	25	"heights_low_4.m4v"	1	1	2
10	26	26	"heights_low_5.m4v"	1	1	1
: more						
1233	3782	3783	"heights_low_2.m4v"	1	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: o	video	video_condition	im_condition	pred_
1	3	3	"heights_low_3.mov"	1	1	2
2	8	8	"heights_low_2.m4v"	1	1	2
3	9	9	"heights_high_1.m4v"	1	2	1
4	10	10	"heights_high_3.m4v"	1	2	1
5	14	14	"heights_high_5.mov"	1	2	2
6	15	15	"heights_high_4.m4v"	1	2	1
7	16	16	"heights_low_6.m4v"	1	1	1
8	20	20	"heights_low_1.m4v"	1	1	2
9	25	25	"heights_low_4.m4v"	1	1	2
10	26	26	"heights_low_5.m4v"	1	1	1
: more						
1233	3782	3783	"heights_low_2.m4v"	1	1	1

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

Correlation Table

```
df3pre1 =
```

	video_scr_mean_miss	video_hp_mean_miss	resp_arousal_mean_miss	resp_fear_mean
1	0.0870833	0.926	0.746667	0.3025
2	0.0823333	0.770727	0.46925	0.405167
3	0.09525	0.86325	0.751917	0.363167
4	0.1095	0.826818	0.693833	0.335
5	0.05225	1.058	0.573167	0.536
6	NaN	NaN	0.628667	0.508917
7	0.0646667	0.841875	0.603167	0.428417
8	0.0953333	0.982182	0.608667	0.0295
9	0.00425	0.674083	0.599583	0.552
10	0.0380833	0.9221	0.394	0.022
: more				
105	0.0721	0.9992	0.3003	0.3003

```
1 df3pre1 = @pipe dfpre[dfpre[:, :video_condition] .== 1, :] |>
2   groupby(_, :participant_num) |>
3   combine(_, [:video_scr, :video_hp, :resp_arousal, :resp_fear, :resp_valence]
4 .=> mean_miss)[:,2:end]
```

```

df3vars1 = 

heights_video_scr_mean_miss  heights_video_hp_mean_miss  heights_resp_arousal_m

1  0.0870833      0.926      0.746667
2  0.0823333      0.770727    0.46925
3  0.09525      0.86325      0.751917
4  0.1095      0.826818    0.693833
5  0.05225      1.058      0.573167
6  NaN          NaN      0.628667
7  0.0646667      0.841875    0.603167
8  0.0953333      0.982182    0.608667
9  0.00425      0.674083    0.599583
10 0.0380833      0.9221     0.394
: more
105 0.0721      0.9992     0.3003

```

```
1 df3vars1 = rename(df3pre1, names(df3pre1) .|> (x -> string("heights_", x)))
```

```
df3pre2 =
```

	video_scr_mean_miss	video_hp_mean_miss	resp_arousal_mean_miss	resp_fear_mean
1	0.00416667	0.923917	0.654667	0.190667
2	0.0169167	0.795083	0.484417	0.37125
3	0.0479167	0.881083	0.622083	0.105167
4	0.0754545	0.8284	0.709636	0.432364
5	0.0306667	1.07033	0.513083	0.465
6	NaN	NaN	0.620667	0.338417
7	0.0519167	0.805222	0.4815	0.300833
8	0.0343333	0.981083	0.42325	0.0935833
9	0.0	0.672333	0.261417	0.235583
10	0.0301667	0.937455	0.23075	0.0065
: more				
105	0.0170833	1.00075	0.316	0.137917

```
1 df3pre2 = @pipe dfpre[dfpre[:, :video_condition] .== 2, :] |>
2   groupby(_, :participant_num) |>
3   combine(_, [:video_scr, :video_hp, :resp_arousal, :resp_fear, :resp_valence]
4 .=> mean_miss)[:, 2:end]
```

```
df3vars2 =
```

	social_video_scr_mean_miss	social_video_hp_mean_miss	social_resp_arousal_mean_r
1	0.00416667	0.923917	0.654667
2	0.0169167	0.795083	0.484417
3	0.0479167	0.881083	0.622083
4	0.0754545	0.8284	0.709636
5	0.0306667	1.07033	0.513083
6	NaN	NaN	0.620667
7	0.0519167	0.805222	0.4815
8	0.0343333	0.981083	0.42325
9	0.0	0.672333	0.261417
10	0.0301667	0.937455	0.23075
⋮ more			
105	0.0170833	1.00075	0.316

```
1 df3vars2 = rename(df3pre2, names(df3pre2) .|> (x -> string("social_", x)))
```

```
df3pre3 =
```

	video_scr_mean_miss	video_hp_mean_miss	resp_arousal_mean_miss	resp_fear_mean
1	0.0518333	0.933667	0.723667	0.236917
2	0.02575	0.793333	0.619833	0.617417
3	0.0560833	0.845083	0.706	0.602333
4	0.0330909	0.840273	0.445364	0.370727
5	0.0129167	1.06758	0.658	0.58675
6	NaN	NaN	0.623917	0.574417
7	0.0561667	0.795667	0.584083	0.44625
8	0.0174167	0.999083	0.490667	0.0255833
9	0.00441667	0.680667	0.40975	0.424917
10	0.0560833	0.9561	0.780083	0.0979167
: more				
105	0.00472727	0.978273	0.398455	0.303

```
1 df3pre3 = @pipe dfpre[dfpre[:, :video_condition] .== 3, :] |>
2   groupby(_, :participant_num) |>
3   combine(_, [:video_scr, :video_hp, :resp_arousal, :resp_fear, :resp_valence]
4 .=> mean_miss)[:,2:end]
```

```
df3vars3 =
```

	spider_video_scr_mean_miss	spider_video_hp_mean_miss	spider_resp_arousal_mean_
--	----------------------------	---------------------------	---------------------------

1	0.0518333	0.933667	0.723667
2	0.02575	0.793333	0.619833
3	0.0560833	0.845083	0.706
4	0.0330909	0.840273	0.445364
5	0.0129167	1.06758	0.658
6	NaN	NaN	0.623917
7	0.0561667	0.795667	0.584083
8	0.0174167	0.999083	0.490667
9	0.00441667	0.680667	0.40975
10	0.0560833	0.9561	0.780083
: more			
105	0.00472727	0.978273	0.398455

```
1 df3vars3 = rename(df3pre3, names(df3pre3) .|> (x -> string("spider_", x)))
```

```
df3 =
```

	participant_num	heights_phobia_mean_miss	social_phobia_mean_miss	spider_phob
1	107	1.7	1.0	1.27778
2	108	2.7	1.66667	5.0
3	109	2.25	1.75	1.77778
4	110	2.75	2.125	1.66667
5	112	3.2	1.70833	2.88889
6	114	2.25	1.83333	4.66667
7	115	1.95	1.83333	2.38889
8	118	1.7	1.25	1.11111
9	119	2.85	1.91667	1.44444
10	120	1.3	1.70833	1.0
: more				
105	232	1.75	2.125	3.72222

```
1 df3 = @pipe dfpre |>
2     groupby(_, :participant_num) |>
3     combine(_, [:heights_phobia, :social_phobia, :spider_phobia] .=> mean_miss)
4 # , [:video_scr, :video_hp, :resp_arousal, :resp_fear, :resp_valence] .=>
5 std_miss
```

```
df3_full =
```

	heights_phobia_mean_miss	social_phobia_mean_miss	spider_phobia_mean_miss	h
1	1.7	1.0	1.27778	0
2	2.7	1.66667	5.0	0
3	2.25	1.75	1.77778	0
4	2.75	2.125	1.66667	0
5	3.2	1.70833	2.88889	0
6	2.25	1.83333	4.66667	N
7	1.95	1.83333	2.38889	0
8	1.7	1.25	1.11111	0
9	2.85	1.91667	1.44444	0
10	1.3	1.70833	1.0	0
: more				
105	1.75	2.125	3.72222	0

```
1 df3_full = reduce(hcat, [df3, df3vars1, df3vars2, df3vars3])[:,2:end]
```

```
corcolnames =
```

```
► ["heights_phobia", "social_phobia", "spider_phobia", "heights_video_scr", "heights_vide
```

```
1 corcolnames = names(df3_full) .|> (x -> x[1:end-10])
```

```
cordf =
```

	heights_phobia	social_phobia	spider_phobia	heights_video_scr	heights_video_hp
--	----------------	---------------	---------------	-------------------	------------------

1	1.0	0.472967	0.330559	0.0376566	-0.162993
2	0.472967	1.0	0.258182	-0.169022	-0.0471457
3	0.330559	0.258182	1.0	0.0399074	-0.174782
4	0.0376566	-0.169022	0.0399074	1.0	-0.0843856
5	-0.162993	-0.0471457	-0.174782	-0.0843856	1.0
6	0.0114751	0.0400155	-0.0545066	0.185075	0.115039
7	0.508414	0.238204	0.257732	-0.0242095	-0.131866
8	-0.461138	-0.213251	-0.233241	-0.0110733	0.102062
9	0.161447	-0.0560544	-0.0366723	0.62288	-0.104923
10	-0.193344	-0.0772595	-0.117803	-0.096999	0.979998
: more					
18	-0.101348	-0.000431601	-0.364054	-0.187217	0.119312

```
1 cordf = DataFrame(nancor(Matrix(df3_full)), corcolnames)
```

```
1 CSV.write("phobiacordf.csv", cordf)
```

	participant_num	heights_phobia_mean_miss	video_scr_mean_miss	video_hp_mean
1	107	1.7	0.0476944	0.927861
2	108	2.7	0.0416667	0.786829
3	109	2.25	0.0664167	0.863139
4	110	2.75	0.0737647	0.831937
5	112	3.2	0.0319444	1.06551
6	114	2.25	NaN	NaN
7	115	1.95	0.0575833	0.813192
8	118	1.7	0.0490278	0.9876
9	119	2.85	0.00288889	0.675694
10	120	1.3	0.0414444	0.938516
: more				
105	232	1.75	0.0296364	0.992788

```

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

```

```

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

```

```
cordfextra =
```

	heights_phobia	video_scr	video_hp	resp_arousal	resp_fear	resp_valence
1	1.0	0.141901	-0.183633	0.0910963	0.498703	-0.315461
2	0.141901	1.0	-0.183235	0.142326	0.0258892	-0.0571948
3	-0.183633	-0.183235	1.0	0.0407244	-0.154579	0.141462
4	0.0910963	0.142326	0.0407244	1.0	0.378368	-0.0900944
5	0.498703	0.0258892	-0.154579	0.378368	1.0	-0.61302
6	-0.315461	-0.0571948	0.141462	-0.0900944	-0.61302	1.0

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

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

Test Area

Column1	Unnamed: o		video	video_condition	im_condition	pred_
1	3	3	"heights_low_3.mov"	1	1	2
2	8	8	"heights_low_2.m4v"	1	1	2
3	9	9	"heights_high_1.m4v"	1	2	1
4	10	10	"heights_high_3.m4v"	1	2	1
5	14	14	"heights_high_5.mov"	1	2	2
6	15	15	"heights_high_4.m4v"	1	2	1
7	16	16	"heights_low_6.m4v"	1	1	1
8	20	20	"heights_low_1.m4v"	1	1	2
9	25	25	"heights_low_4.m4v"	1	1	2
10	26	26	"heights_low_5.m4v"	1	1	1
: more						
1233	3782	3783	"heights_low_2.m4v"	1	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 "heights_phobia",
22 "participant_num_str"];
```

```
dfcleancol =
```

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_rt
1	"heights_low_3.mov"	107	0.048	2.503	0.045
2	"heights_low_2.m4v"	107	0.033	3.754	0.036
3	"heights_high_1.m4v"	107	0.032	2.52	0.024
4	"heights_high_3.m4v"	107	0.039	2.219	0.036
5	"heights_high_5.mov"	107	0.058	3.754	0.053
6	"heights_high_4.m4v"	107	0.282	4.355	0.17
7	"heights_low_6.m4v"	107	0.287	1.485	0.321
8	"heights_low_1.m4v"	107	0.05	2.319	0.044
9	"heights_low_4.m4v"	107	0.289	1.802	0.038
10	"heights_low_5.m4v"	107	0.059	1.368	0.056
: more					
1233	"heights_low_2.m4v"	232	0.264	4.722	0.081

```
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 "heights_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_an
1	"heights_high_2.mov"	224	0.513	1.102	0.5
2	"heights_low_3.mov"	224	0.503	7.075	0.5
3	"heights_high_6.mov"	224	0.5	0.935	0.5
4	"heights_low_2.m4v"	224	0.5	0.484	0.5
5	"heights_high_1.m4v"	224	0.496	2.87	0.5
6	"heights_high_3.m4v"	224	0.5	0.684	0.5
7	"heights_low_1.m4v"	224	0.534	5.356	0.5
8	"heights_low_6.m4v"	224	0.5	1.418	0.5
9	"heights_high_5.mov"	224	0.486	9.411	0.5
10	"heights_high_4.m4v"	224	0.5	2.203	0.5
11	"heights_low_5.m4v"	224	0.5	1.652	0.5
12	"heights_low_4.m4v"	224	0.5	0.651	0.5

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

```
▶ [0.0, 0.053, 0.103, 0.102, 0.158, 0.156, 0.118, 0.05, 0.05, 0.0, 0.154, 0.101, 0.0, 0.1
```

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

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

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

```
-0.012923105957940008
```

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

```
0.038559033641762615
```

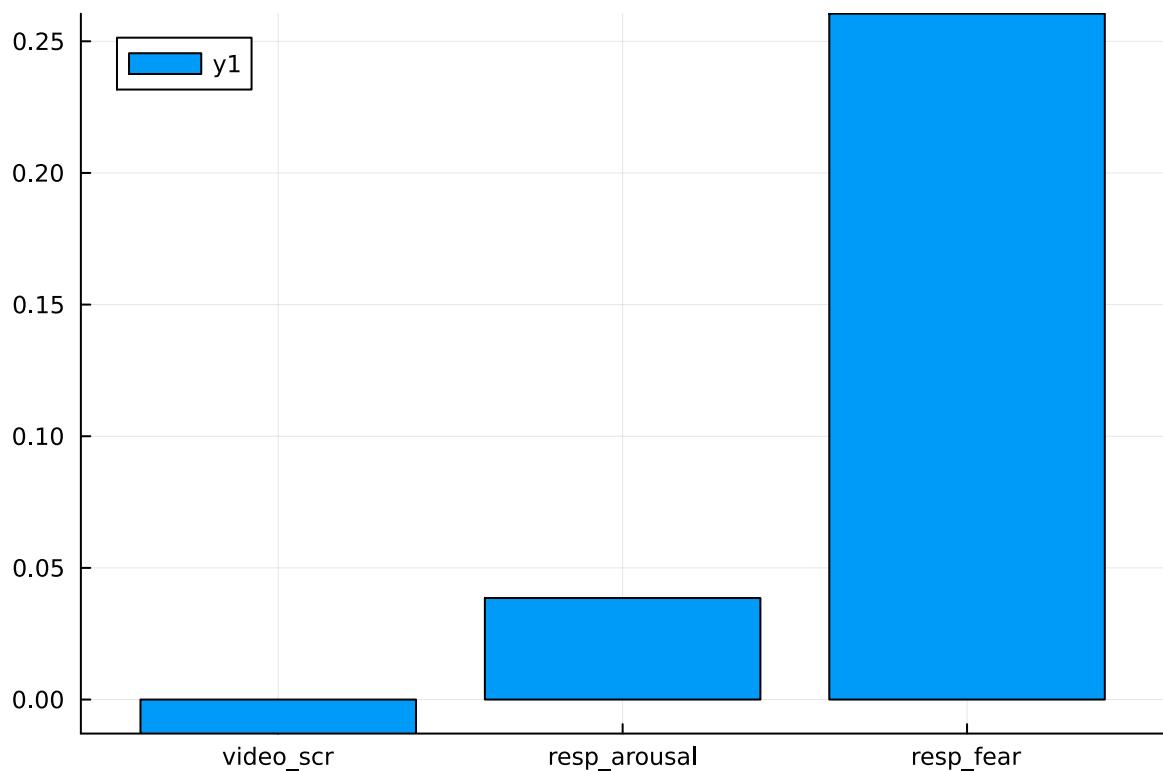
```
1 comparevar("resp_arousal")
```

```
0.2604838550544422
```

```
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.DensePred...
video_scr ~ 1 + resp_arousal
```

Coefficients:

	Coef.	Std. Error	t	Pr(> t)	Lower 95%	Upper 95%
(Intercept)	0.0402875	0.0037051	10.87	<1e-25	0.0330185	0.0475565
resp_arousal	0.0499454	0.00607627	8.22	<1e-15	0.0380244	0.0618664

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

0.04028753141479971

```
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_a
	String31	Int64	Float64	Float64	Float64
1	heights_low_3.mov	107	0.048	2.503	
2	heights_low_2.m4v	107	0.033	3.754	
3	heights_high_1.m4v	107	0.032	2.52	
4	heights_high_3.m4v	107	0.039	2.219	
5	heights_high_5.mov	107	0.058	3.754	
6	heights_high_4.m4v	107	0.282	4.355	
7	heights_low_6.m4v	107	0.287	1.485	
8	heights_low_1.m4v	107	0.05	2.319	
9	heights_low_4.m4v	107	0.289	1.802	
10	heights_low_5.m4v	107	0.059	1.368	
11	heights_high_6.mov	107	0.055	2.419	
12	heights_high_2.mov	107	0.103	1.785	

:

Last Group (10 rows): participant_num_str = "232"

Row	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_a
	String31	Int64	Float64	Float64	Float64
1	heights_high_1.m4v	232	0.352	4.822	
2	heights_low_3.mov	232	0.5	2.303	
3	heights_high_3.m4v	232	0.51	4.138	
4	heights_low_4.m4v	232	0.393	3.454	
5	heights_high_6.mov	232	0.35	4.388	
6	heights_low_6.m4v	232	0.295	55.528	
7	heights_low_5.m4v	232	0.308	11.947	
8	heights_high_2.mov	232	0.236	8.943	
9	heights_high_4.m4v	232	0.306	7.008	
10	heights_low_2.m4v	232	0.264	4.722	

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

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_an
1	"heights_high_6.mov"	114	0.752	2.503	0.611
2	"heights_high_4.m4v"	114	0.79	2.503	0.709
3	"heights_low_6.m4v"	114	0.665	2.92	0.482
4	"heights_low_3.mov"	114	0.554	2.62	0.617
5	"heights_low_1.m4v"	114	0.448	2.436	0.33
6	"heights_high_5.mov"	114	0.5	2.186	0.5
7	"heights_low_2.m4v"	114	0.736	2.169	0.761
8	"heights_high_2.mov"	114	0.706	1.819	0.729
9	"heights_low_4.m4v"	114	0.5	1.752	0.5
10	"heights_low_5.m4v"	114	0.317	2.887	0.289
11	"heights_high_3.m4v"	114	0.53	2.186	0.5
12	"heights_high_1.m4v"	114	0.51	1.769	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 "heights_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

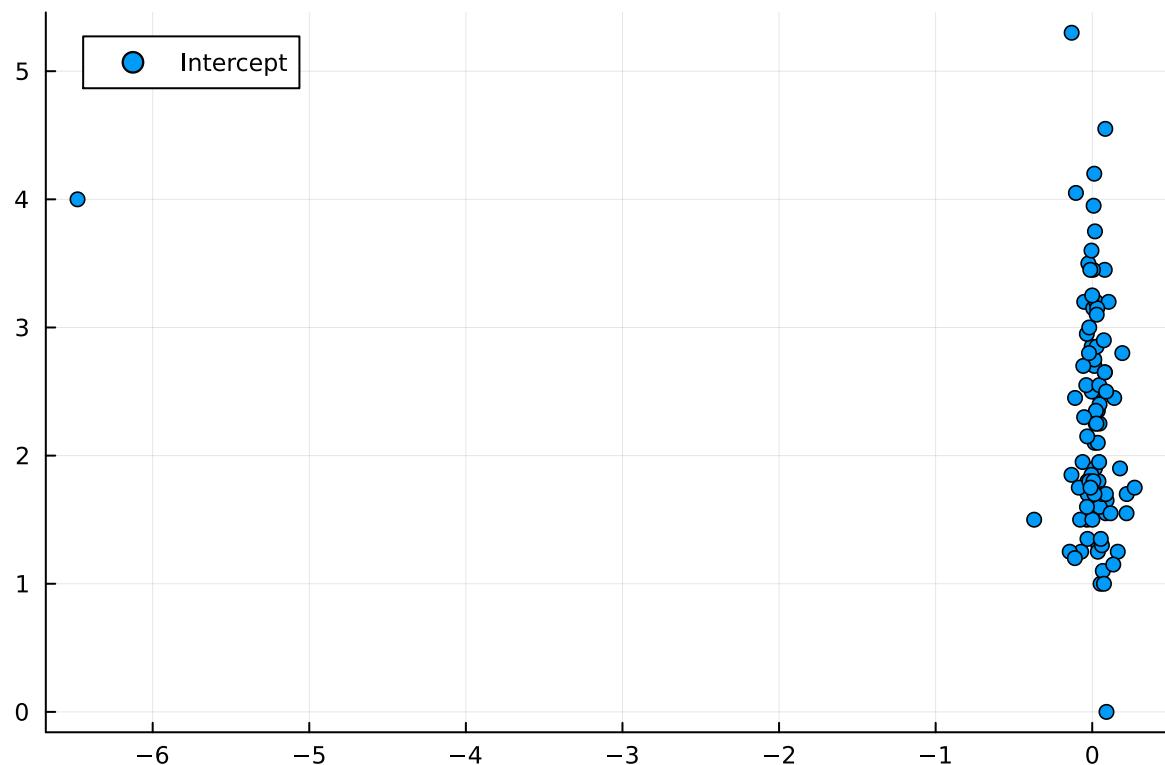
`df2_subjlv1 =`

	participant_num_str	video_scr_resp_arousal	heights_phobia
1	"107"	► (-0.0294172, 0.156027, 20.3867)	1.7
2	"108"	► (0.0141399, 0.145324, 19.3618)	2.7
3	"109"	► (0.0221785, 0.0971803, 18.5038)	2.25
4	"110"	► (0.0133715, 0.138547, 22.0338)	2.75
5	"112"	► (-0.0511336, 0.180373, 20.984)	3.2
6	"115"	► (-0.0615343, 0.209231, 17.4484)	1.95
7	"118"	► (0.220318, -0.205341, 18.3522)	1.7
8	"119"	► (-0.00383224, 0.0134798, 34.83)	2.85
9	"120"	► (0.0426463, -0.0115811, 19.8829)	1.3
10	"122"	► (0.0389395, 0.102618, 20.9955)	1.3
⋮ more			
95	"232"	► (-0.0108559, 0.276243, 22.3012)	1.75

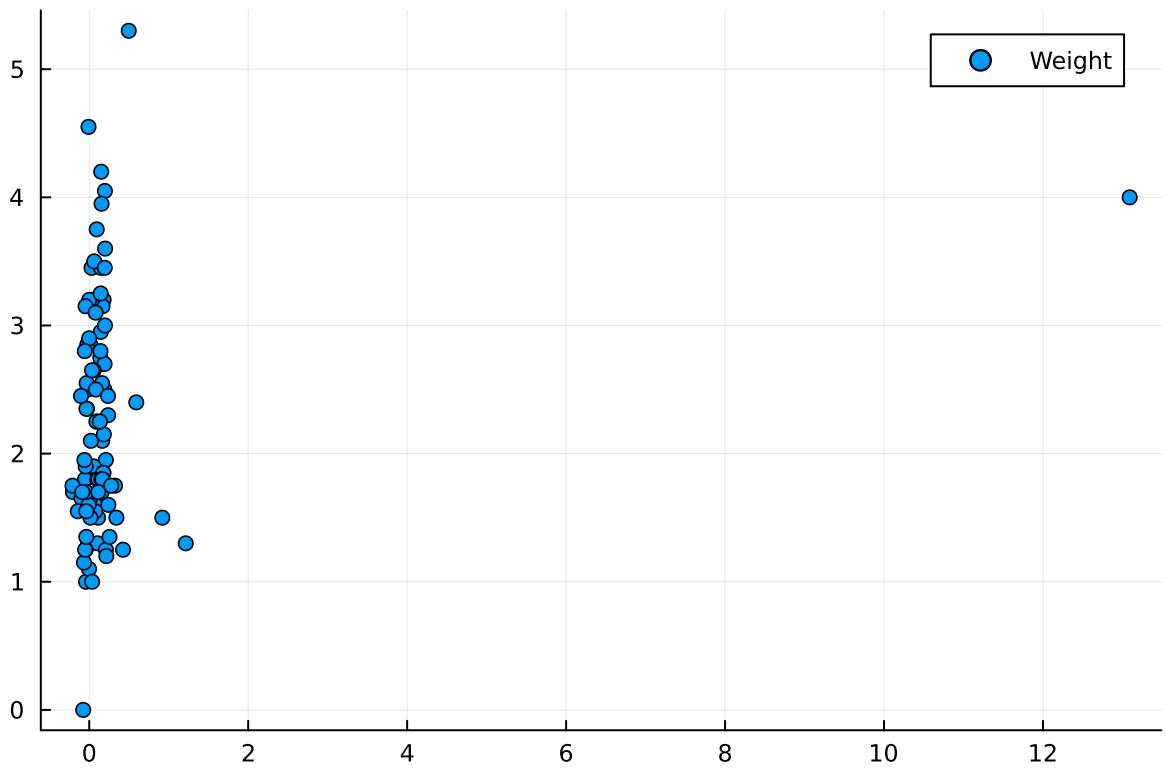
```

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

```



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



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

```
scr_to_aro_int = -0.21189583908938775
```

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

```
scr_to_aro_coef = 0.2042837512528689
```

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

```
▶ [20.3867, 19.3618, 18.5038, 22.0338, 20.984, 17.4484, 18.3522, 34.83, 19.8829, 20.9955]
```

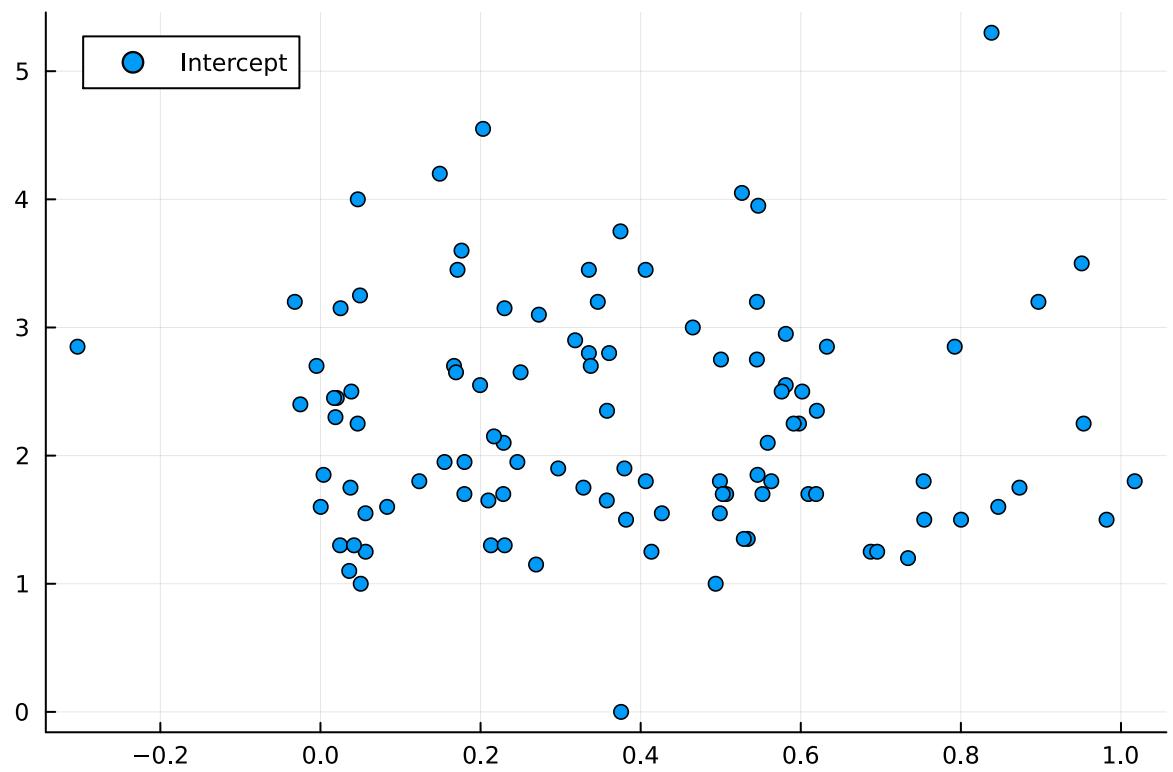
```
1 df2_subjlvl.video_scr_resp_arousal .|> x -> x[3]
```

Fitting Arousal to Fear Response

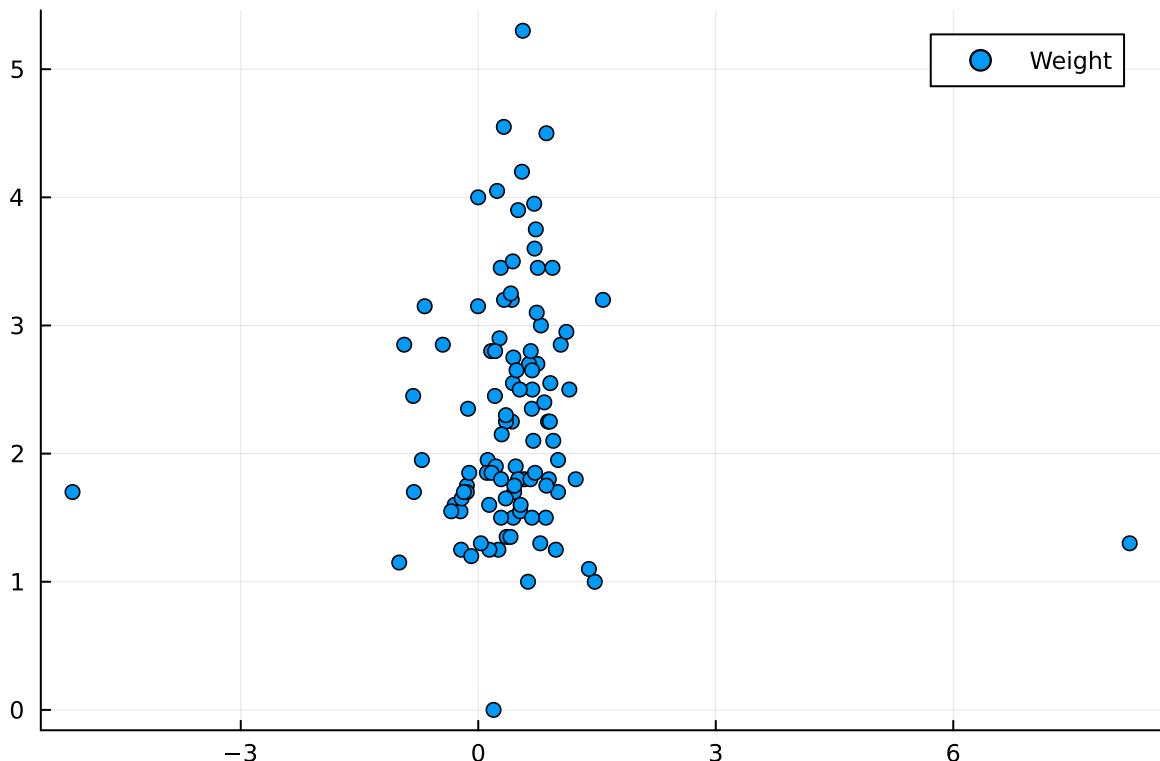
```
df2_subjlvl_2 =
```

	participant_num_str	resp_arousal_resp_fear	heights_phobia
1	"107"	► (0.609566, 0.453225, 5.8691)	1.7
2	"108"	► (0.166877, 0.746293, 6.17599)	2.7
3	"109"	► (0.597806, 0.424352, 1.58945)	2.25
4	"110"	► (0.545174, 0.443759, 0.676331)	2.75
5	"112"	► (0.346466, 0.422949, 11.9338)	3.2
6	"114"	► (0.179951, 0.881707, 12.8563)	2.25
7	"115"	► (0.552143, 0.119097, 12.3195)	1.95
8	"118"	► (0.632647, -0.81288, 17.0034)	1.7
9	"119"	► (0.0245644, 1.0417, 9.90382)	2.85
10	"120"	► (0.212973, 8.22849, -1.17997)	1.3
⋮ more			
104	"232"	► (0.0419014, 0.860468, 7.79749)	1.75

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



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



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

```
aro_to_fear_int = -0.17245110040792372
```

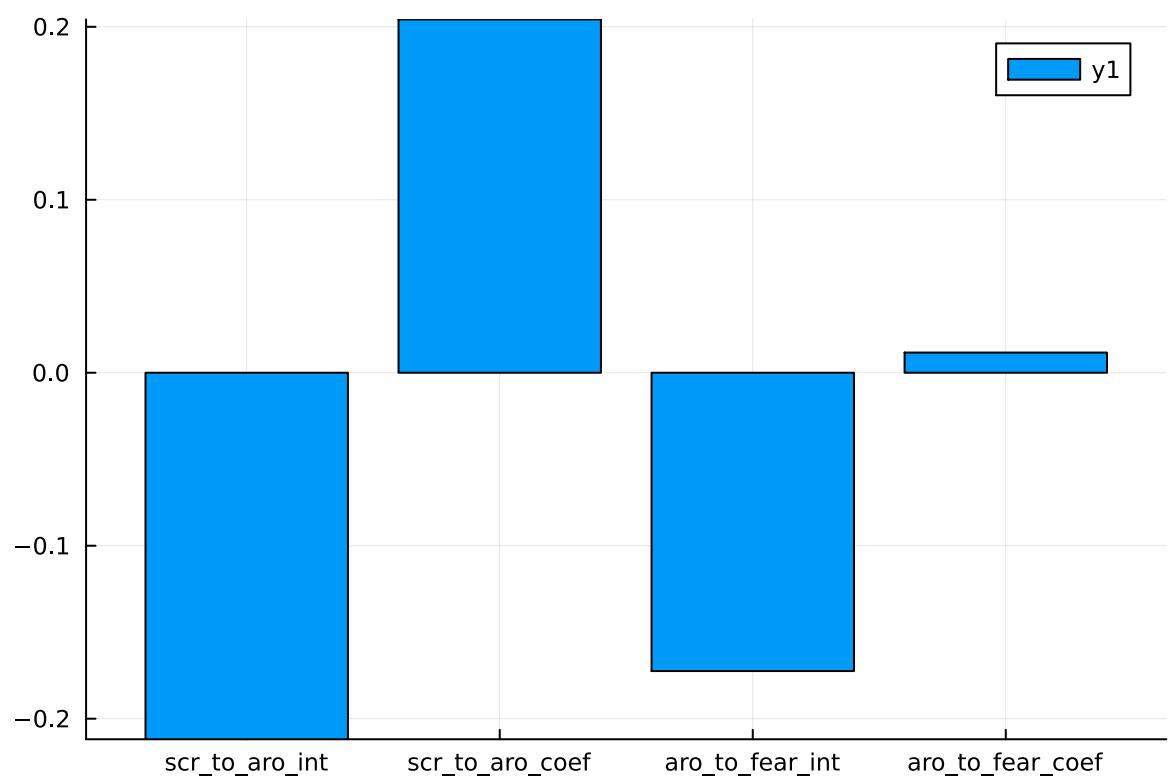
```
1 aro_to_fear_int = cor(df2_subjlv2.resp_arousal_resp_fear .|> x -> x[1],
df2_subjlv2.heights_phobia)
```

```
aro_to_fear_coef = 0.011662764820563476
```

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

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_fear
1	"heights_low_3.mov"	107	0.048	2.503	0.045
2	"heights_low_2.m4v"	107	0.033	3.754	0.036
3	"heights_high_1.m4v"	107	0.032	2.52	0.024
4	"heights_high_3.m4v"	107	0.039	2.219	0.036
5	"heights_high_5.mov"	107	0.058	3.754	0.053
6	"heights_high_4.m4v"	107	0.282	4.355	0.17
7	"heights_low_6.m4v"	107	0.287	1.485	0.321
8	"heights_low_1.m4v"	107	0.05	2.319	0.044
9	"heights_low_4.m4v"	107	0.289	1.802	0.038
10	"heights_low_5.m4v"	107	0.059	1.368	0.056
: more					
1233	"heights_low_2.m4v"	232	0.264	4.722	0.081

1 dfcleancol

mm2 =

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.4623	0.0203	22.78	<1e-99	0.1796
video_scr	1.0109	0.1191	8.49	<1e-16	
Residual	0.2291				

Minimizing 2 Time: 0:00:00 (75.00 ms/it)
 objective: 159.03687385603598

Minimizing 20 Time: 0:00:00 (10.76 ms/it)

	Column	Variance	Std.Dev
participant_num_str	(Intercept)	0.032247	0.179575
Residual		0.052465	0.229053

```
1 VarCorr(mm2)
```

```
▶ [1×105 Matrix{Float64}:
 0.172852 -0.0672146 0.170205 0.106371 ... -0.205337 0.10034 0.227672 -0.20205
1 ranef(mm2)
```

0.028574795465711653

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

	participant_num_str	(Intercept)
1	"107"	0.172852
2	"108"	-0.0672146
3	"109"	0.170205
4	"110"	0.106371
5	"112"	0.0510754
6	"114"	0.146462
7	"115"	0.0664402
8	"118"	0.0439839
9	"119"	0.117068
10	"120"	-0.0940886
:	more	
105	"232"	-0.202058

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

Mixed Models

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

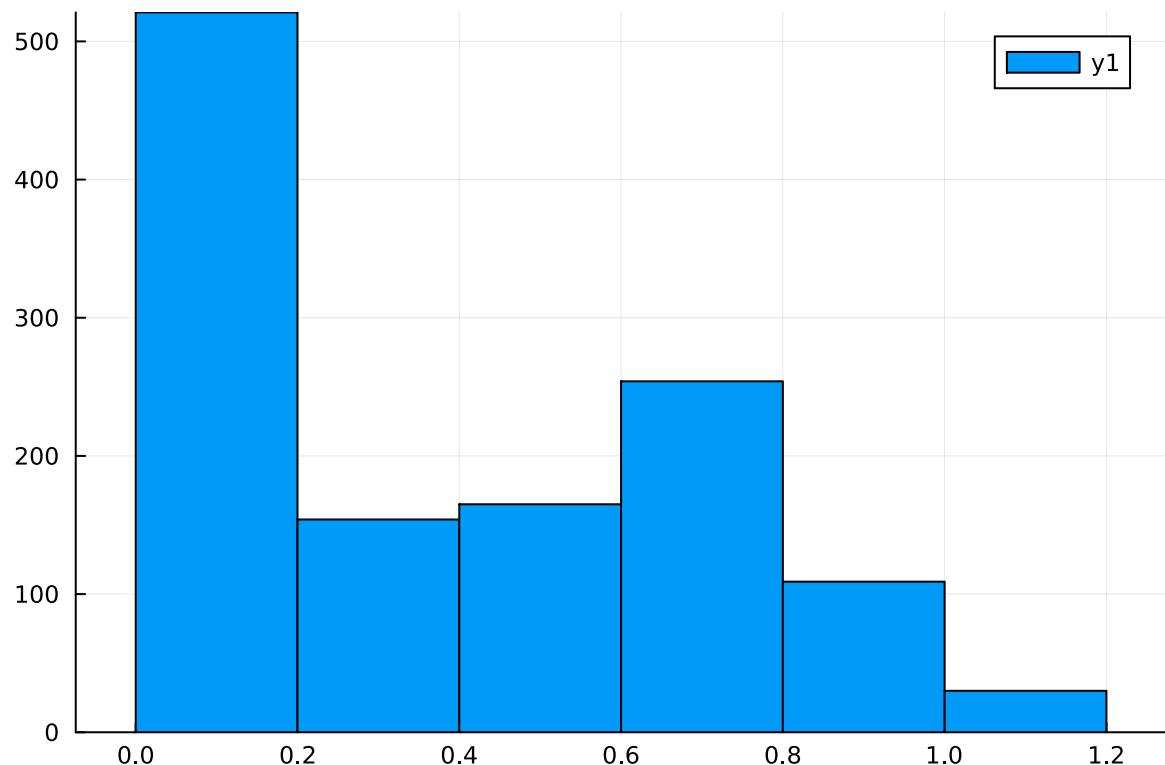
	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_fear
1	"heights_low_3.mov"	107	0.048	2.503	0.045
2	"heights_low_2.m4v"	107	0.033	3.754	0.036
3	"heights_high_1.m4v"	107	0.032	2.52	0.024
4	"heights_high_3.m4v"	107	0.039	2.219	0.036
5	"heights_high_5.mov"	107	0.058	3.754	0.053
6	"heights_high_4.m4v"	107	0.282	4.355	0.17
7	"heights_low_6.m4v"	107	0.287	1.485	0.321
8	"heights_low_1.m4v"	107	0.05	2.319	0.044
9	"heights_low_4.m4v"	107	0.289	1.802	0.038
10	"heights_low_5.m4v"	107	0.059	1.368	0.056
: more					
1233	"heights_low_2.m4v"	232	0.264	4.722	0.081

```

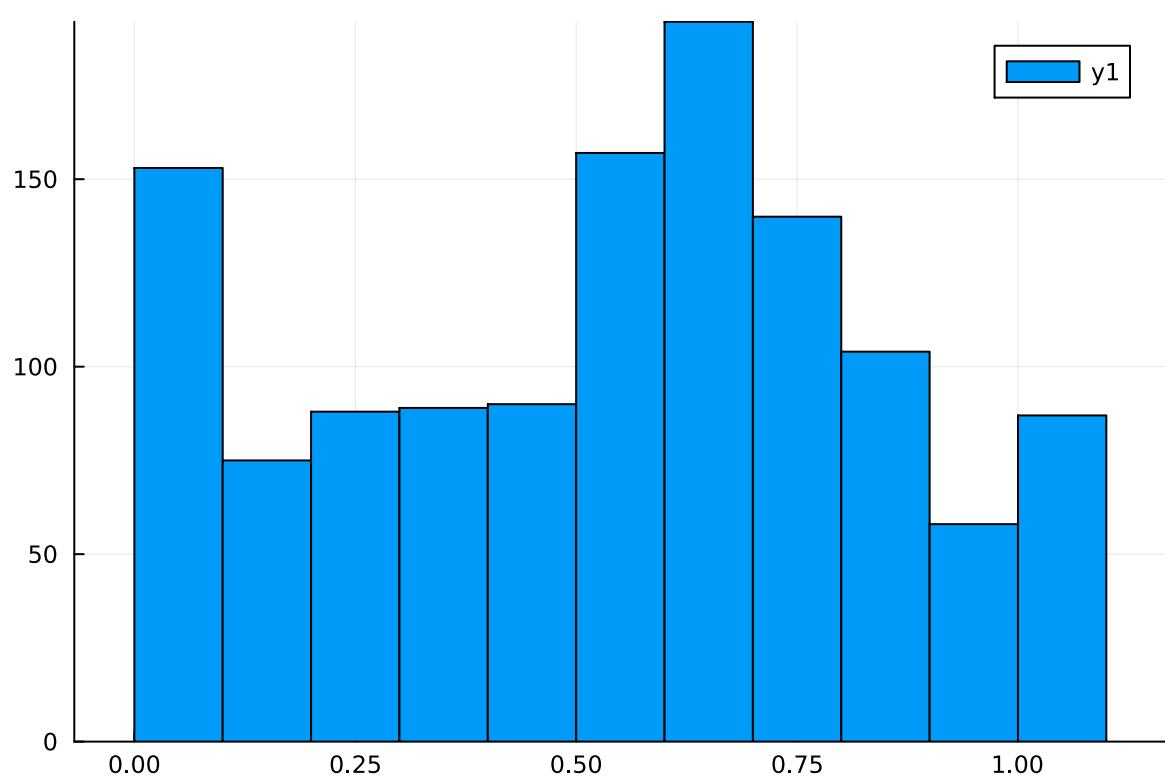
1 dfcleancol2 = @pipe dfcleancol |>
2     groupby(_, :participant_num_str) |>
3     transform(_, [:video_scr, :video_hp, :resp_arousal,
4      :scr_change_video, :resp_valence] .=> mean)|>
5     transform(_, [:video_scr, :video_scr_mean] => ByRow(-) =>
6      :video_scr_cmc,
7      [:video_hp, :video_hp_mean] => 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)	0.4594	0.0369	12.46	<1e-34	0.1806
video_scr_cmc	0.8985	0.1773	5.07	<1e-06	1.2116
video_scr_mean	1.0558	0.4809	2.20	0.0281	
Residual	0.2193				

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

Scr to arousal, loglike: -27.208289501774402

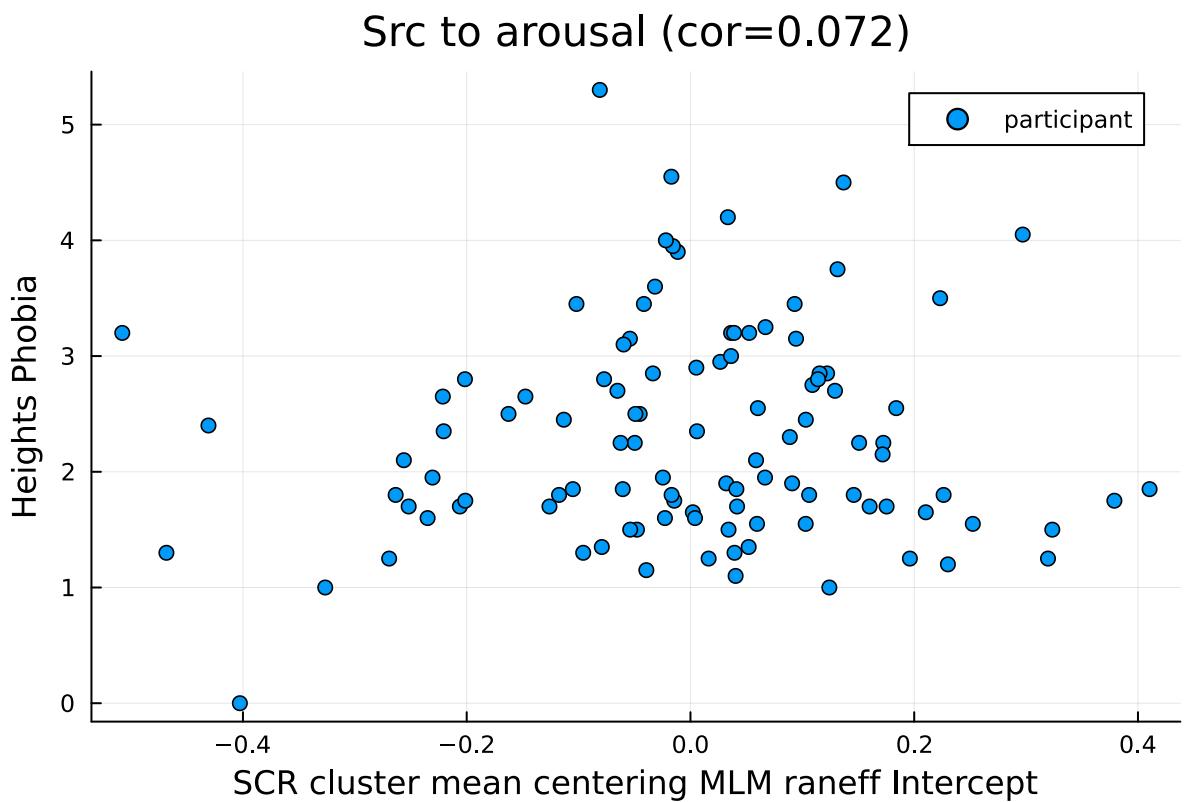
```
1 md"#### Scr to arousal, loglike: $(loglikelihood(mm3))"
```

Scr to arousal, AIC: 68.4165790035488

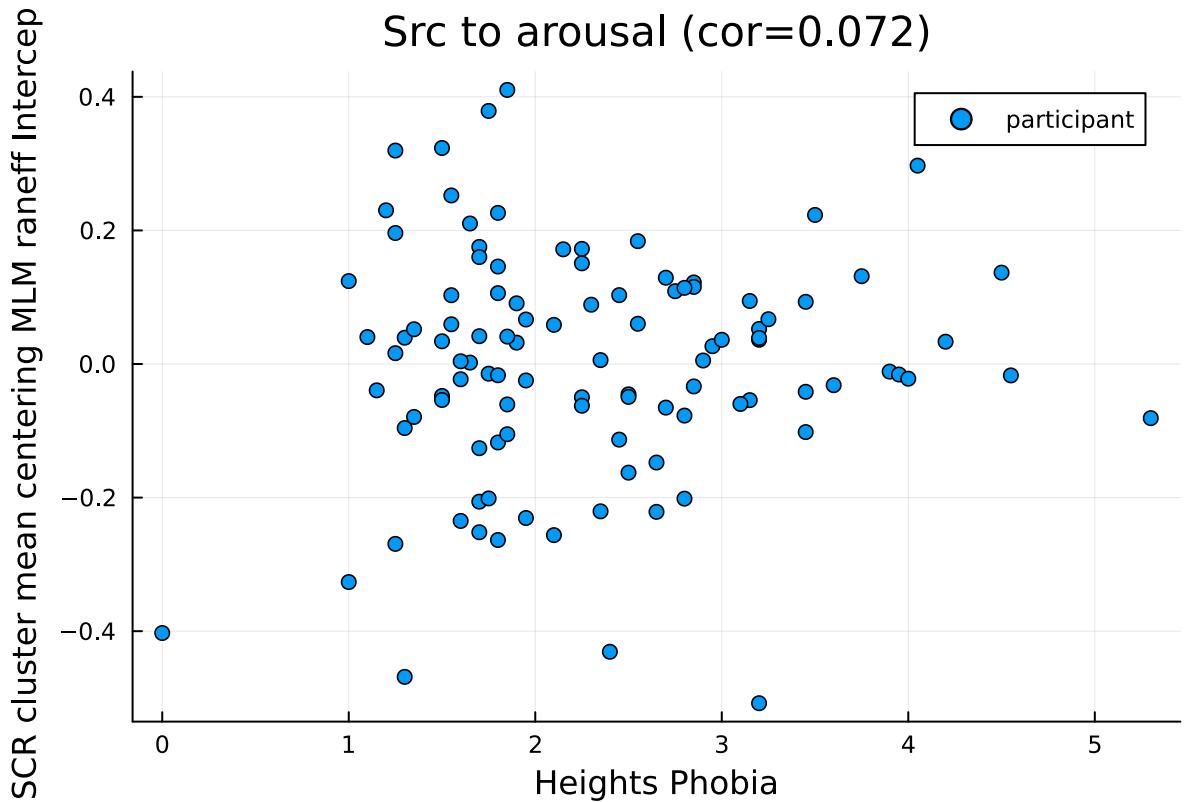
```
1 md"#### Scr to arousal, AIC: $(aic(mm3))"
```

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

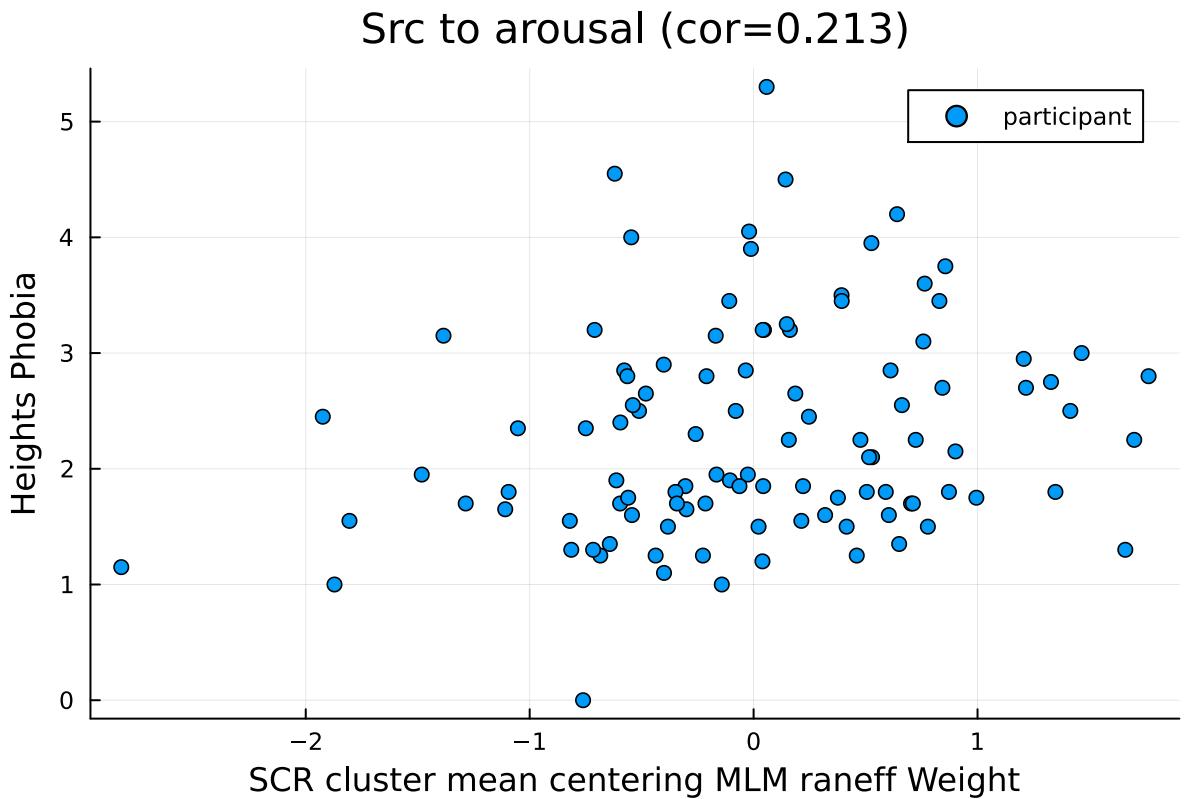
```
1 mm3coef_w = cor(ranef(mm3)[1][2,:], df2_subjlvl_hp.heights_phobia);
```



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



```
1 scatter( df2_subjlv_hp.heights_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="Heights Phobia")
```



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

Model 2: arousal to fear

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

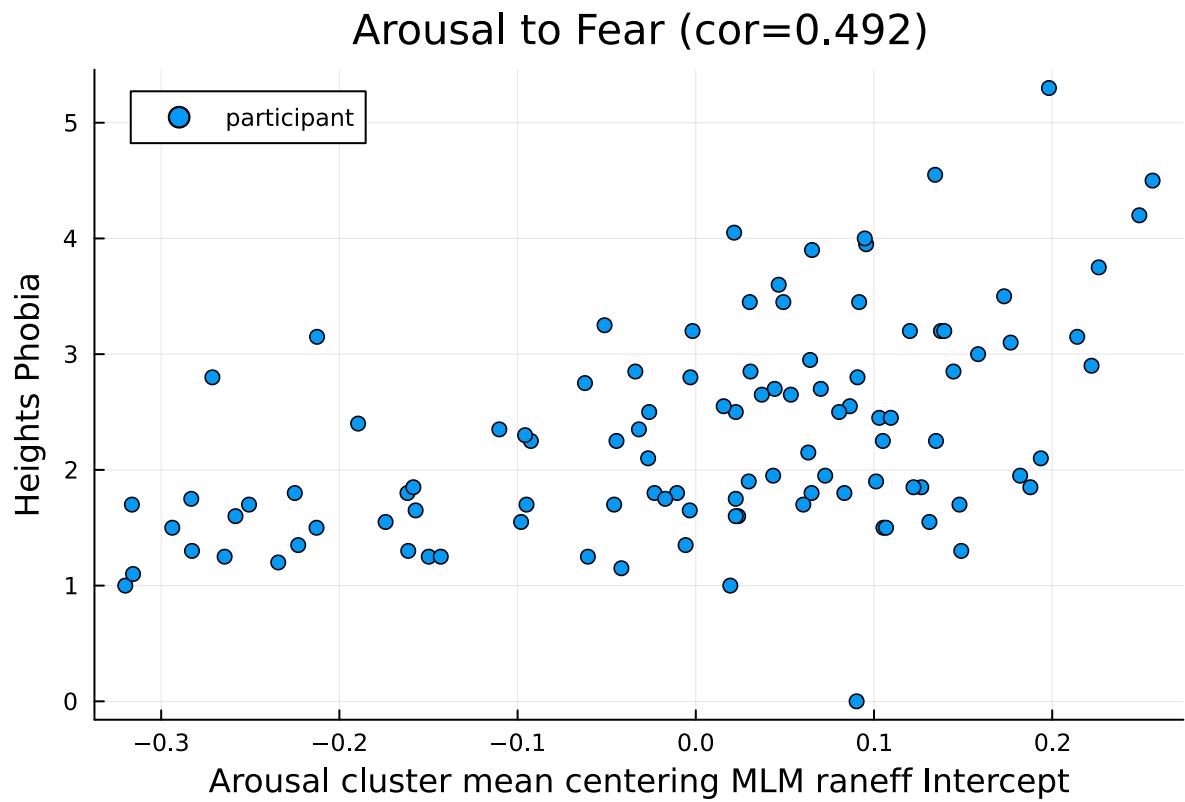
```
mm4 =
```

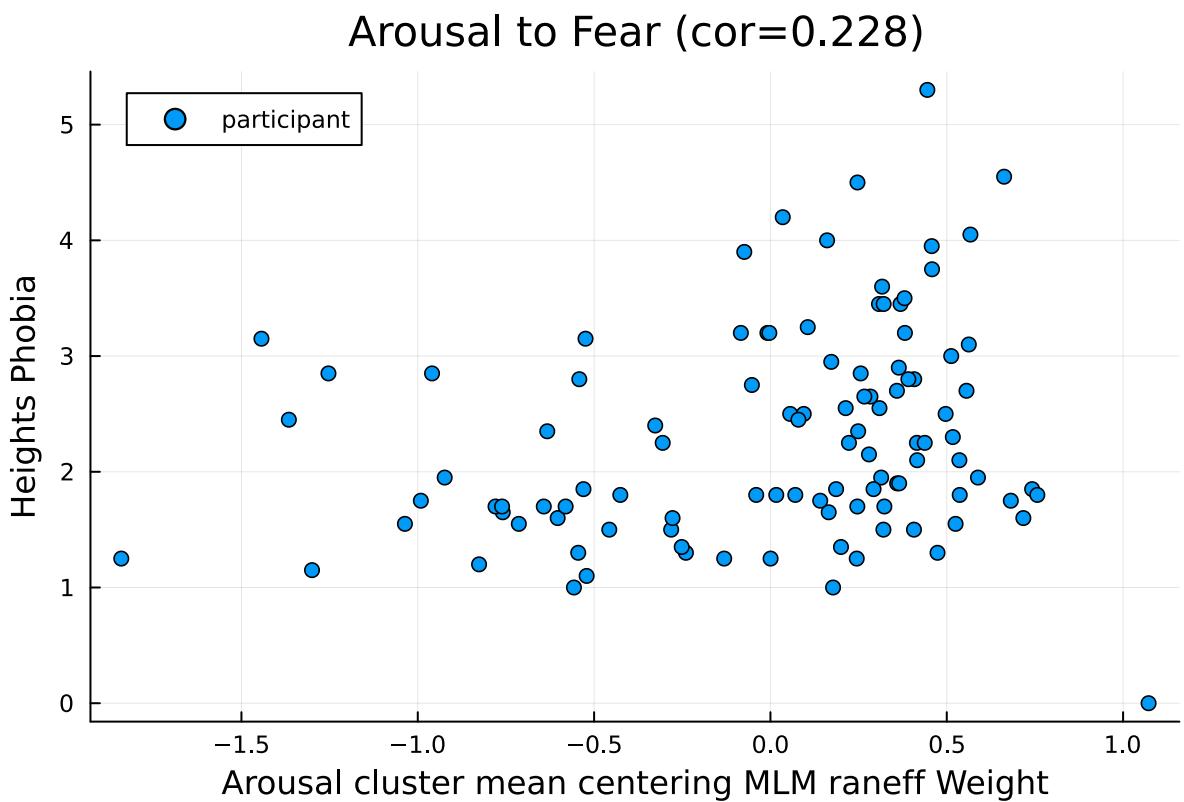
	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.2606	0.0449	5.81	<1e-08	0.1559
resp_arousal_cmc	0.6154	0.0719	8.56	<1e-16	0.6385
resp_arousal_mean	0.2109	0.0790	2.67	0.0076	
Residual	0.2034				

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

Arousal to fear, loglike: 24.980327153317063

Arousal to fear, AIC: -35.960654306634126





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

Model 3: scr to fear

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

```
mm6 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.3895	0.0322	12.11	<1e-33	0.1553
video_scr_cmc	2.3274	0.1552	14.99	<1e-50	0.7213
video_scr_mean	-0.2605	0.4151	-0.63	0.5303	
Residual	0.2508				

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

Scr to fear, loglike: -137.6093450192117

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

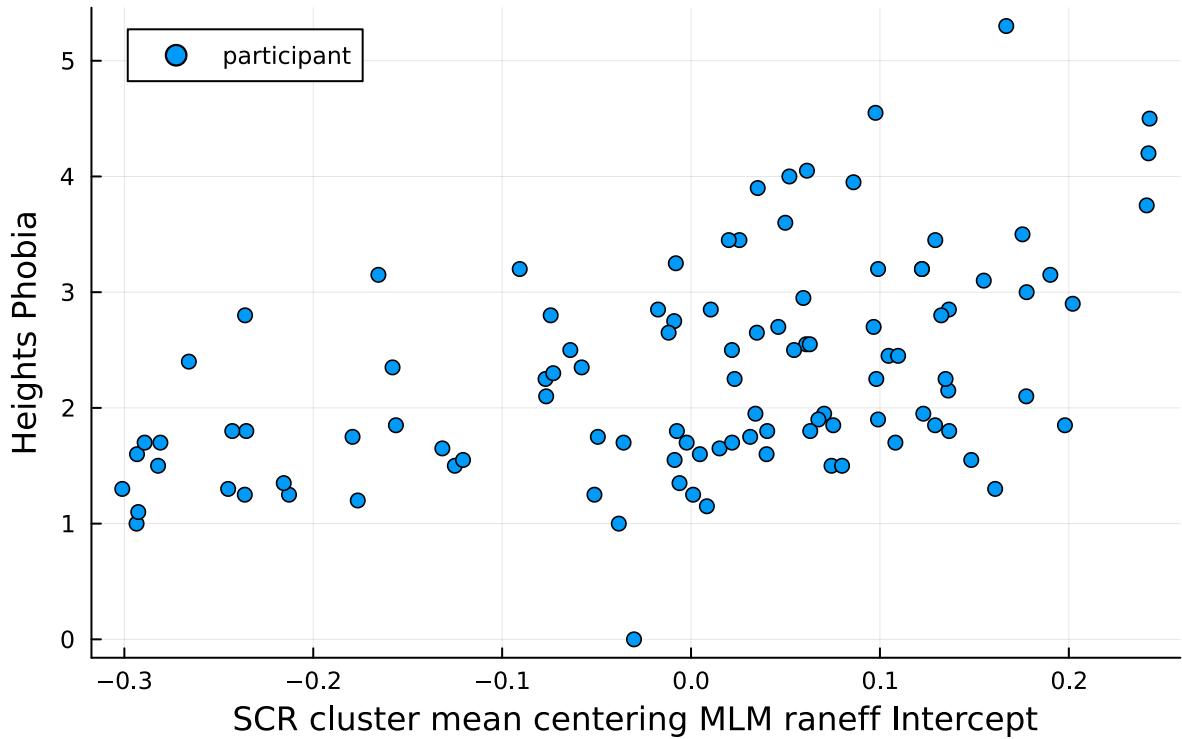
Scr to fear, AIC: 289.2186900384234

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

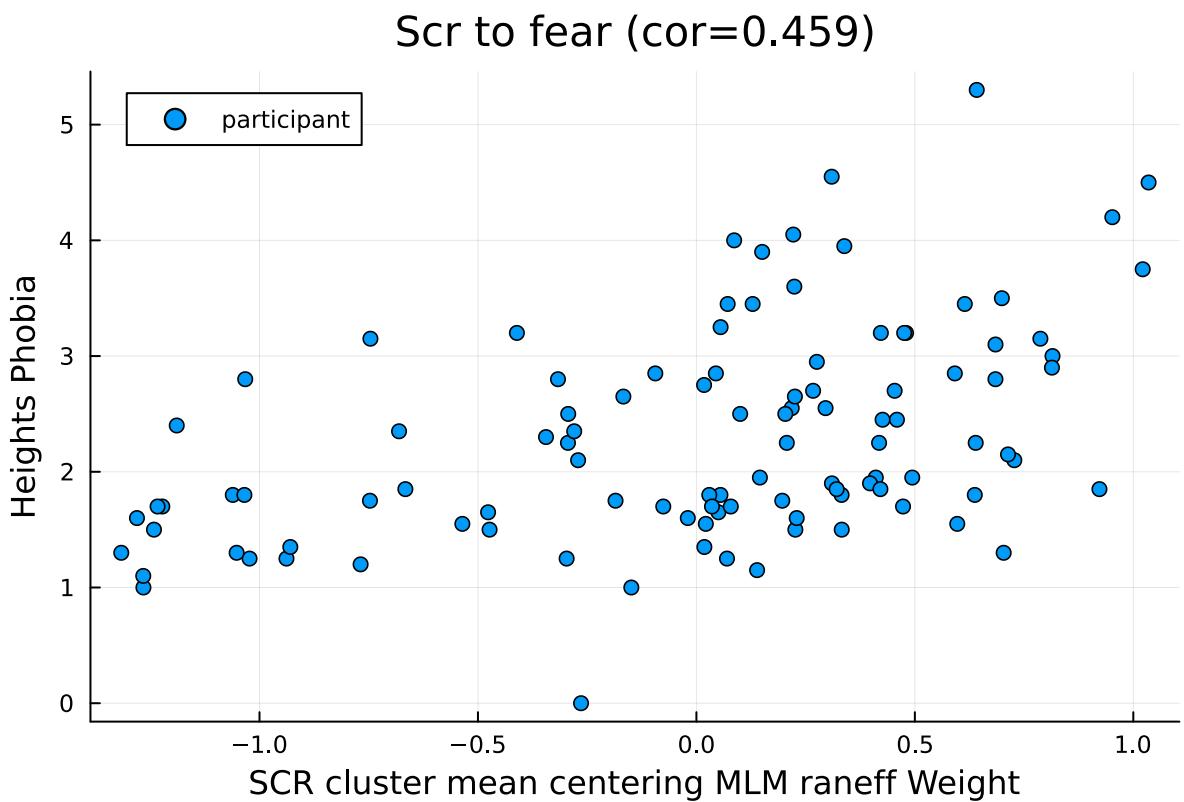
```
1 mm6coef_i = cor(ranef(mm6)[1][1,:], df2_subjlvl_hp.heights_phobia);
```

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

Scr to fear (cor=0.477)



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



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

Model 4: scr to valence

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

```
mm12 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.5762	0.0267	21.60	<1e-99	0.1168
video_scr_cmc	-2.1037	0.1551	-13.57	<1e-41	0.5698
video_scr_mean	0.1935	0.3453	0.56	0.5751	
Residual	0.2654				

```
1 mm12 = fit(LinearMixedModel, @formula(resp_valence ~ video_scr_cmc +
  video_scr_mean + (video_scr_cmc|participant_num_str)), dfcleancol2)
```

Scr to arousal, loglike: -178.58254233849124

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

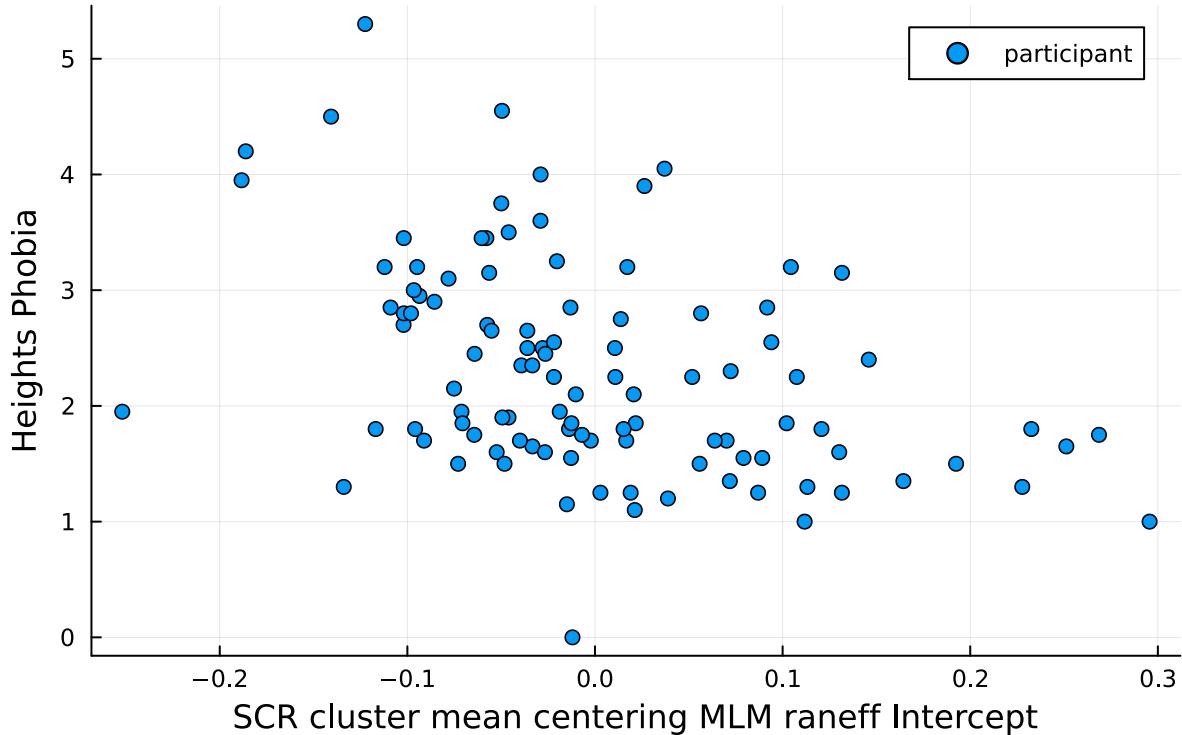
Scr to arousal, AIC: 371.1650846769825

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

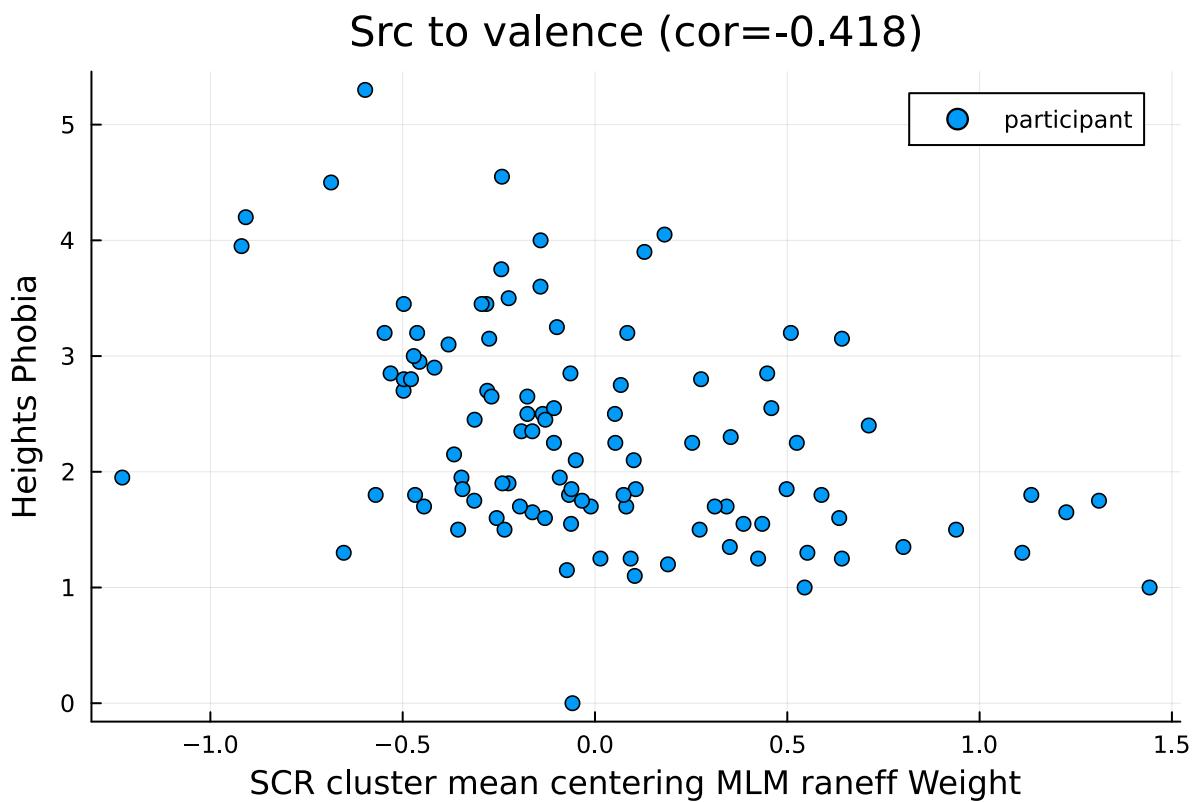
```
1 mm12coef_i = cor(ranef(mm12)[1][1,:], df2_subjlvl_hp.heights_phobia);
```

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

Src to valence (cor=-0.418)



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



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

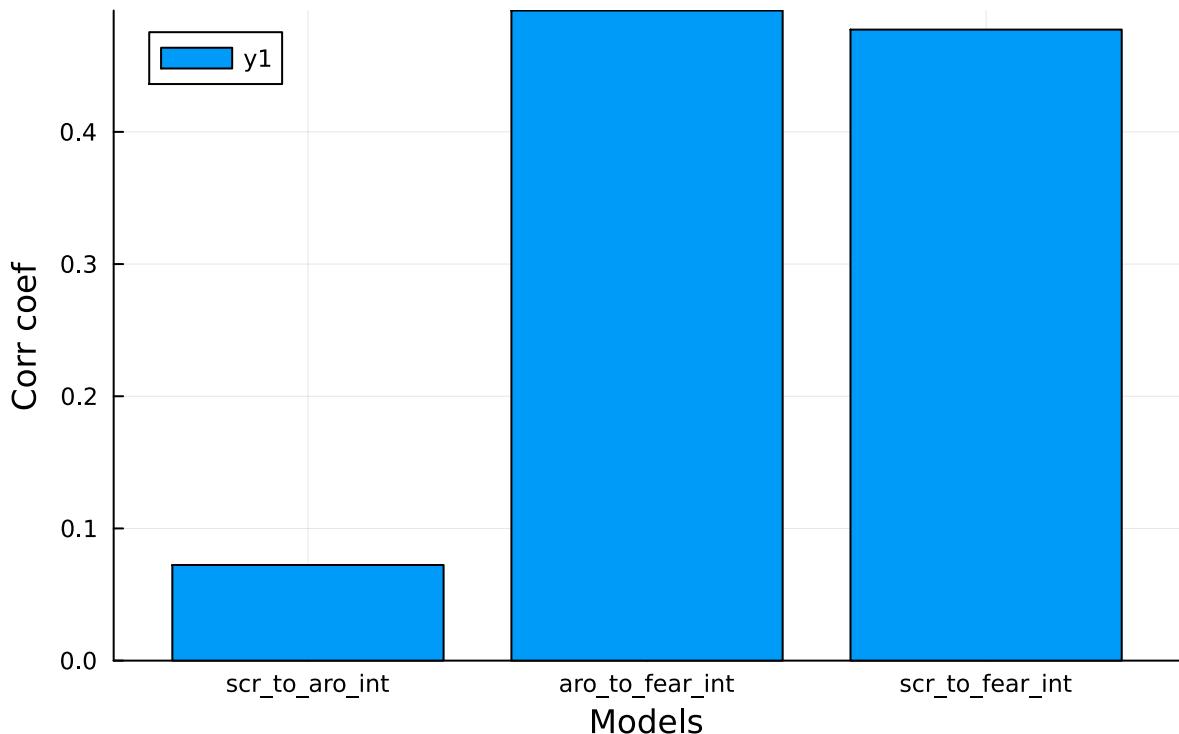
Model correlation comparison plots

```
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"];
```

Intercept term correlations to trait phobia

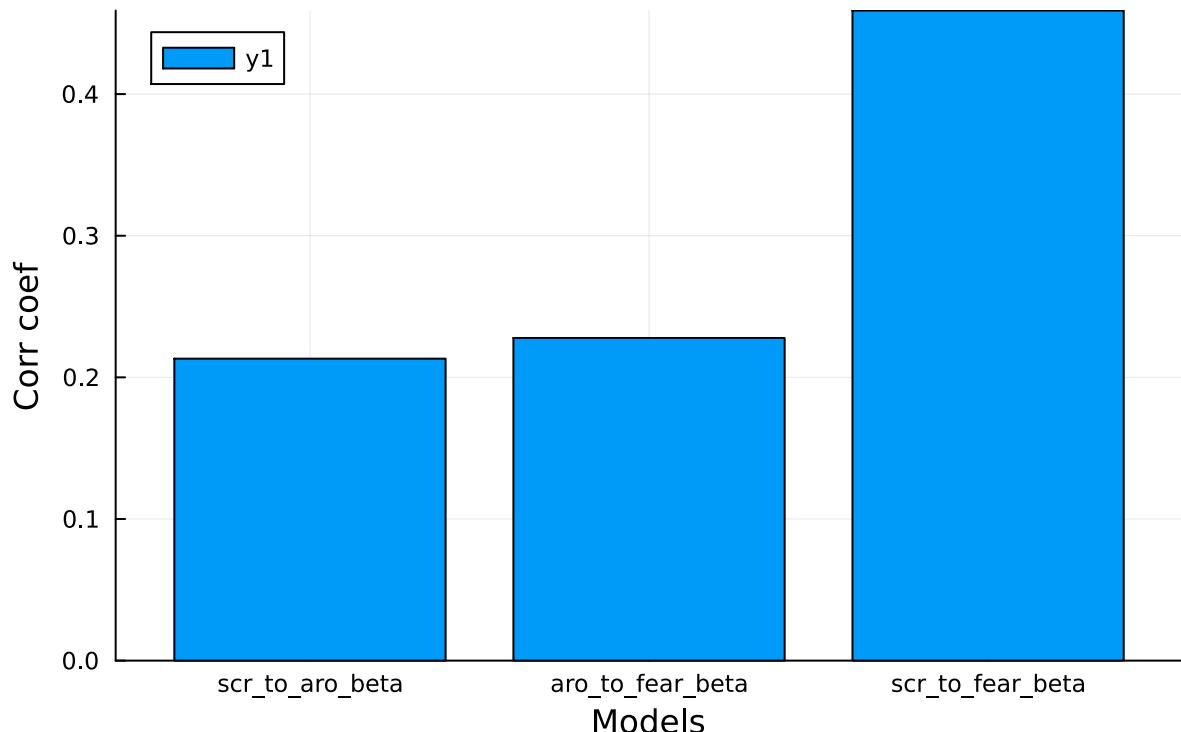


```
1 plot(bar(plotcols3, plotdata3), title="Intercept term correlations to trait phobia", xlabel="Models", ylabel="Corr coef")
```

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

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

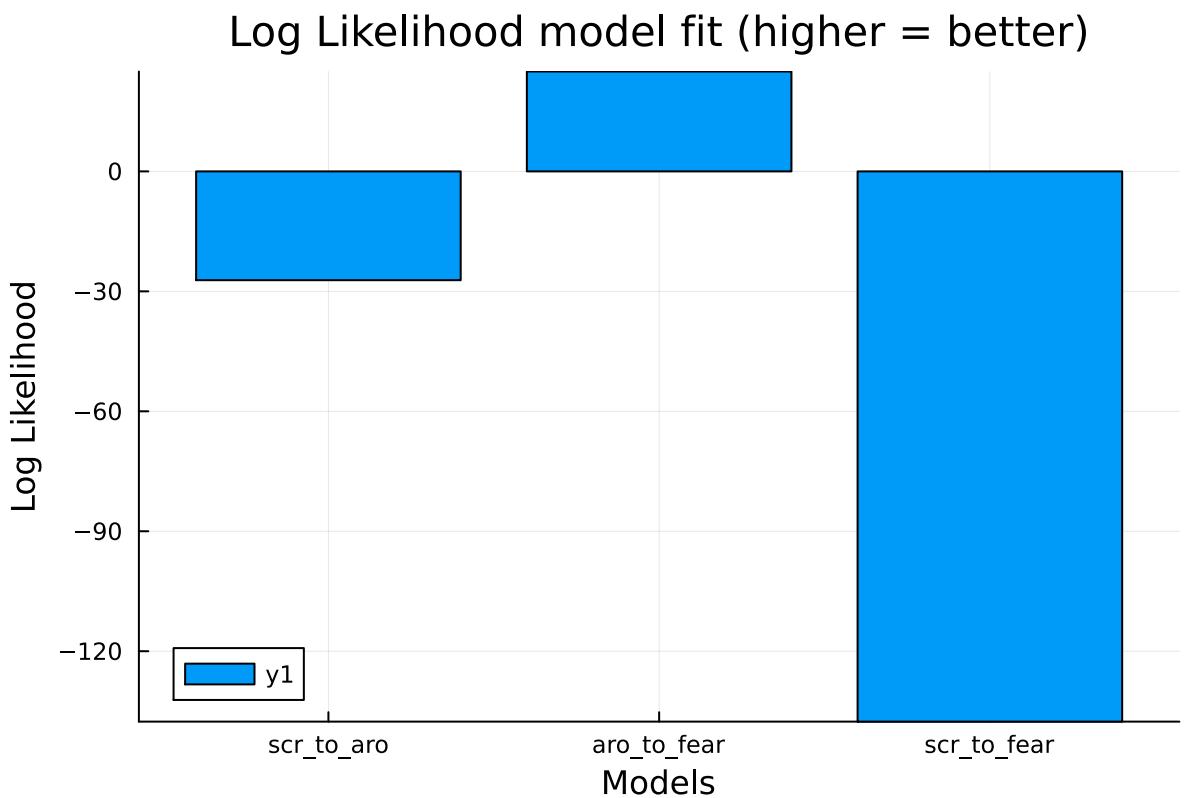
Beta/Weight term correlations to trait phobia



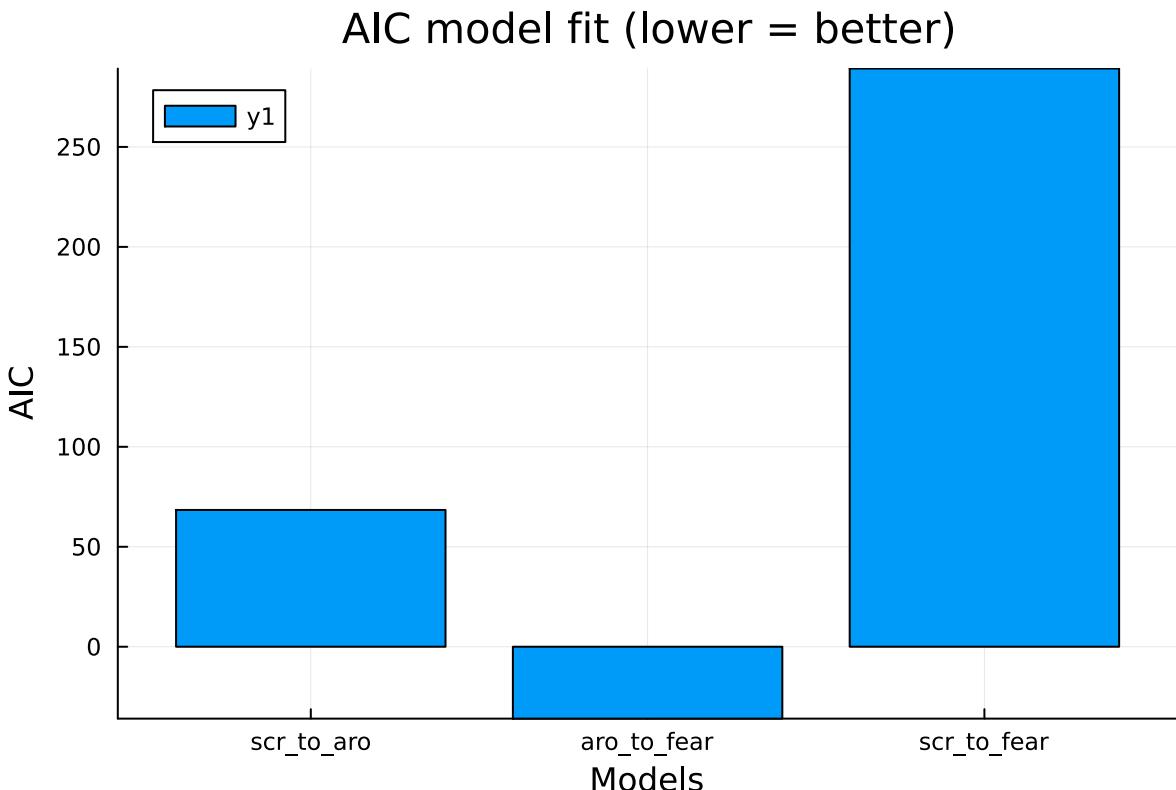
```
1 plot(bar(plotcols4, plotdata4), title="Beta/Weight term correlations to trait phobia", xlabel="Models", ylabel="Corr coef")
```

```
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 =
```

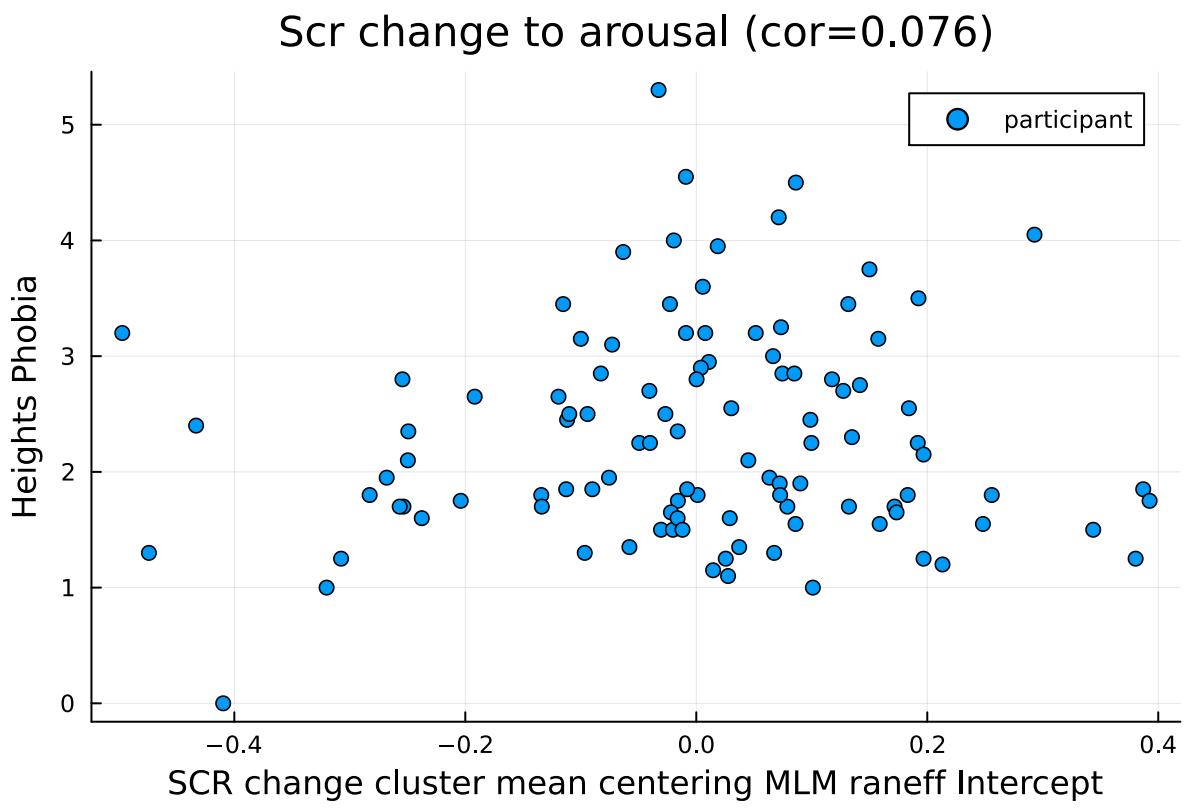
	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.5173	0.0238	21.69	<1e-99	0.1847
scr_change_video_cmc	0.8820	0.1797	4.91	<1e-06	1.2303
scr_change_video_mean	0.4667	0.5647	0.83	0.4085	
Residual	0.2193				

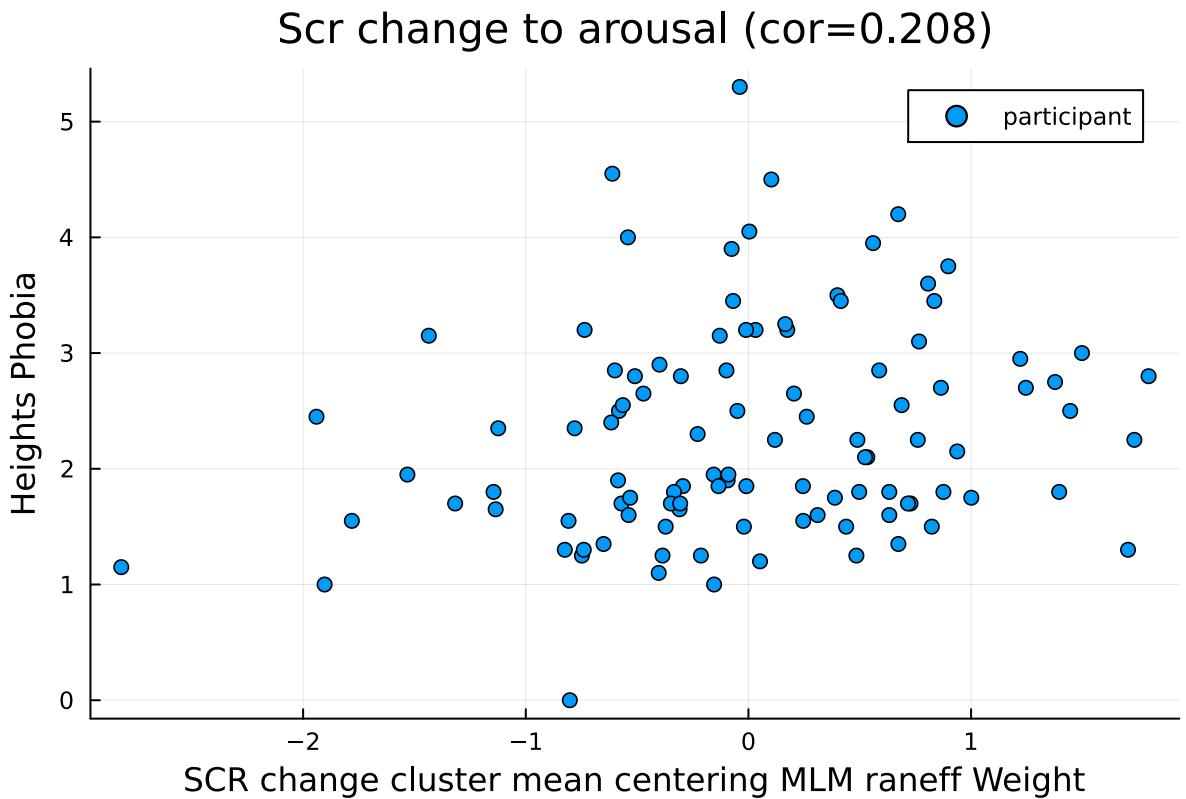
```
1 mm5 = fit(LinearMixedModel, @formula(resp_arousal ~ scr_change_video_cmc +  
      scr_change_video_mean + (scr_change_video_cmc|participant_num_str)), dfcleancol2)
```

-29.25065078031804

```
1 loglikelihood(mm5)
```

72.50130156063608





Scr change to Fear

```
mm7 =
```

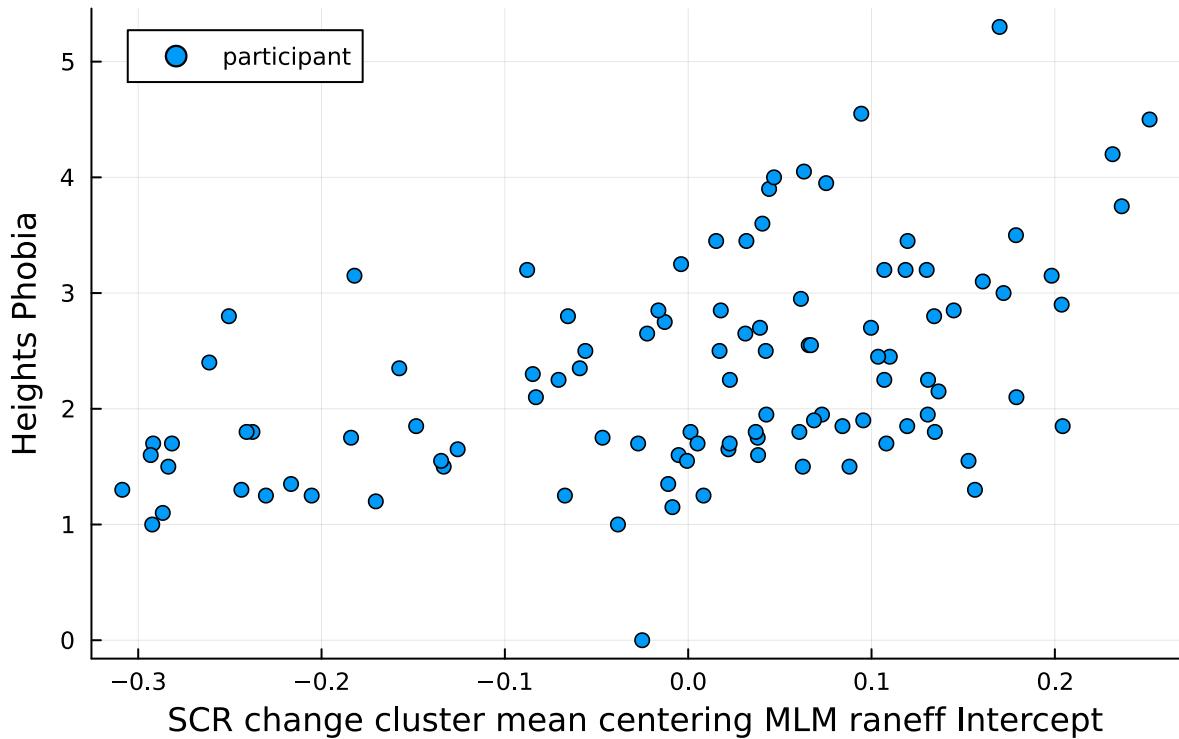
	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.3788	0.0204	18.53	<1e-75	0.1558
scr_change_video_cmc	2.3523	0.1552	15.15	<1e-51	0.7343
scr_change_video_mean	-0.2576	0.4594	-0.56	0.5751	
Residual	0.2508				

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

-137.78765454448893

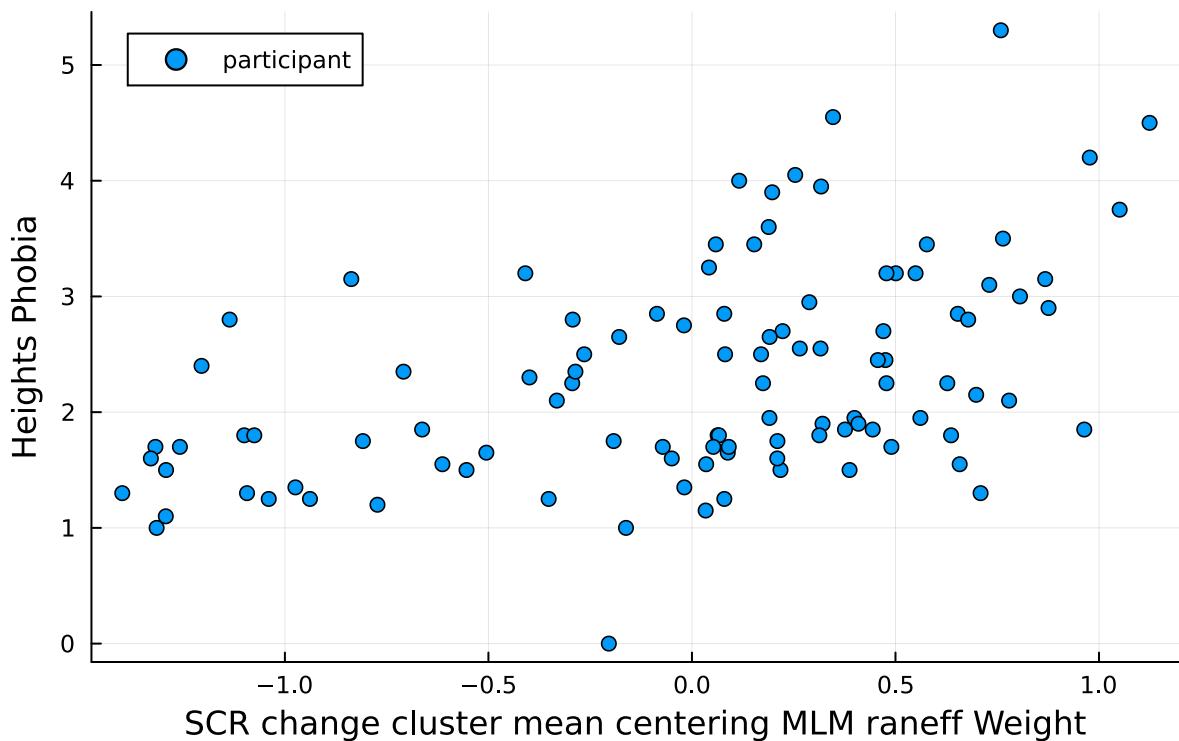
```
1 mm7coef_i = cor(ranef(mm7)[1][1,:], df2_subjlvl_hp.heights_phobia);
```

Scr change to fear (cor=0.473)



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

Scr change to fear (cor=0.464)



Scr to Arousal with heights phobia modulating term

```
mm8 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.4247	0.0602	7.05	<1e-11	0.1801
video_scr_cmc	-0.0220	0.4625	-0.05	0.9621	1.1613
heights_phobia	0.0147	0.0204	0.72	0.4728	
video_scr_mean	1.0701	0.4800	2.23	0.0258	
video_scr_cmc & heights_phobia	0.4054	0.1878	2.16	0.0309	
Residual	0.2193				

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

```
-24.789538551311736
```

```
1 loglikelihood(mm8)
```

```
67.57907710262347
```

```
1 aic(mm8)
```

```
mm8coef_i = 9.216959943853414e-15
```

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

```
mm8coef_w = 8.755549605129113e-16
```

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

```
mm9 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.0663	0.0518	1.28	0.2006	0.1325
resp_arousal_cmc	0.2898	0.1994	1.45	0.1460	0.6247
heights_phobia	0.0901	0.0155	5.80	<1e-08	
resp_arousal_mean	0.1862	0.0698	2.67	0.0076	
resp_arousal_cmc & heights_phobia	0.1415	0.0812	1.74	0.0812	
Residual	0.2034				

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

```
39.59030744504807
```

```
1 loglikelihood(mm9)
```

```
mm10 =
```

	Est.	SE	z	p	σ_participant_num_str
(Intercept)	0.1740	0.0466	3.73	0.0002	0.1304
video_scr_cmc	2.0525	0.4129	4.97	<1e-06	0.7175
heights_phobia	0.0923	0.0160	5.78	<1e-08	
video_scr_mean	-0.2135	0.3603	-0.59	0.5535	
video_scr_cmc & heights_phobia	0.1192	0.1660	0.72	0.4727	
Residual	0.2507				

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

```
-121.89308625320413
```

```
1 loglikelihood(mm10)
```

```
mm11 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.0620	0.0528	1.17	0.2407	0.1352
video_scr_cmc	1.6754	0.3684	4.55	<1e-05	0.7442
heights_phobia	0.0874	0.0157	5.56	<1e-07	
resp_arousal_cmc	0.2596	0.1838	1.41	0.1578	0.5668
video_scr_mean	-0.4967	0.3446	-1.44	0.1495	
resp_arousal_mean	0.2576	0.0668	3.86	0.0001	
video_scr_cmc & heights_phobia	-0.1288	0.1504	-0.86	0.3918	
video_scr_cmc & resp_arousal_cmc	2.0879	1.7785	1.17	0.2404	2.5301
heights_phobia & resp_arousal_cmc	0.1195	0.0750	1.59	0.1111	
video_scr_cmc & heights_phobia & resp_arousal_cmc	-0.1720	0.7407	-0.23	0.8164	
Residual	0.1857				

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

```
> Minimizing 1247    Time: 0:00:00 (80.19 µs/it)  
objective: -260.2071657107716
```

```
Minimizing 1414    Time: 0:00:00 (77.79 µs/it)
```

```
Minimizing 1415    Time: 0:00:00 (77.74 µs/it)  
objective: -260.2074005105393
```

```
mm13 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.5217	0.0735	7.10	<1e-11	0.1269
heights_phobia	0.0561	0.0155	3.62	0.0003	
video_scr_cmc	0.8207	0.2142	3.83	0.0001	0.0590
resp_arousal_cmc	0.2671	0.0929	2.88	0.0040	0.1976
resp_valence_cmc	-0.6053	0.0800	-7.56	<1e-13	0.2262
video_scr_mean	-0.2194	0.2583	-0.85	0.3955	
resp_arousal_mean	0.1546	0.0518	2.99	0.0028	
resp_valence_mean	-0.5808	0.0813	-7.15	<1e-12	
heights_phobia & video_scr_cmc	-0.0544	0.0888	-0.61	0.5403	
heights_phobia & resp_arousal_cmc	0.0313	0.0392	0.80	0.4246	
video_scr_cmc & resp_arousal_cmc	-0.2810	1.4116	-0.20	0.8422	2.0613
heights_phobia & resp_valence_cmc	-0.0019	0.0334	-0.06	0.9556	
video_scr_cmc & resp_valence_cmc	-1.4774	0.9860	-1.50	0.1341	1.4076
resp_arousal_cmc & resp_valence_cmc	-0.4315	0.3145	-1.37	0.1701	0.6180
heights_phobia & video_scr_cmc & resp_arousal_cmc	0.6242	0.6167	1.01	0.3115	
heights_phobia & video_scr_cmc & resp_valence_cmc	0.9275	0.4161	2.23	0.0258	
heights_phobia & resp_arousal_cmc & resp_valence_cmc	0.1060	0.1279	0.83	0.4073	
video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	4.1541	3.5510	1.17	0.2421	1.4796
heights_phobia & video_scr_cmc & resp_arousal_cmc & resp_valence_cmc	-0.1827	1.5677	-0.12	0.9072	
Residual	0.1268				

```
1 mm13= fit(LinearMixedModel, @formula(resp_fear ~ heights_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 572 Time: 0:00:00 (0.18 ms/it)



```
objective: -1159.5063115162839

Minimizing 1269    Time: 0:00:00 ( 0.16 ms/it)
objective: -1163.4820625641908

Minimizing 1960    Time: 0:00:00 ( 0.16 ms/it)
objective: -1163.8624961596463

Minimizing 2651    Time: 0:00:00 ( 0.16 ms/it)
objective: -1164.1527947647983

Minimizing 3343    Time: 0:00:00 ( 0.16 ms/it)
objective: -1164.3402660556503

Minimizing 4034    Time: 0:00:00 ( 0.16 ms/it)
objective: -1164.4689708093697

Minimizing 4725    Time: 0:00:00 ( 0.16 ms/it)
objective: -1164.5474838187251
```

130.10370025526964

```
1 loglikelihood(mm11)
```

582.308122868242

```
1 loglikelihood(mm13)
```

-218.2074005105393

```
1 aic(mm11)
```

-1052.616245736484

```
1 aic(mm13)
```

mm14 =

	Est.	SE	z	p	σ_participant_num_str
(Intercept)	0.5287	0.0718	7.36	<1e-12	0.1272
heights_phobia	0.0567	0.0153	3.71	0.0002	
video_scr_cmc	0.7927	0.2797	2.83	0.0046	0.6143
video_hp_cmc	0.1346	0.0724	1.86	0.0632	0.0838
resp_arousal_cmc	0.2730	0.0979	2.79	0.0053	0.2311
resp_valence_cmc	-0.5273	0.0826	-6.38	<1e-09	0.2433
video_scr_mean	-0.0342	0.2513	-0.14	0.8919	
video_hp_mean	-0.0561	0.0325	-1.73	0.0845	
resp_arousal_mean	0.1830	0.0487	3.76	0.0002	
resp_valence_mean	-0.5744	0.0766	-7.50	<1e-13	
heights_phobia & video_scr_cmc	-0.0558	0.1162	-0.48	0.6314	
heights_phobia & video_hp_cmc	-0.0154	0.0301	-0.51	0.6101	
video_scr_cmc & video_hp_cmc	0.3960	1.8161	0.22	0.8274	2.8498
heights_phobia & resp_arousal_cmc	0.0234	0.0407	0.58	0.5651	
video_scr_cmc & resp_arousal_cmc	0.1451	1.4799	0.10	0.9219	2.4603
video_hp_cmc & resp_arousal_cmc	0.4229	0.3997	1.06	0.2900	0.1240
heights_phobia & resp_valence_cmc	-0.0284	0.0341	-0.83	0.4048	
video_scr_cmc & resp_valence_cmc	-0.7263	0.9757	-0.74	0.4567	1.3719
video_hp_cmc & resp_valence_cmc	-0.0148	0.3102	-0.05	0.9619	0.4508
resp_arousal_cmc & resp_valence_cmc	-0.5432	0.2778	-1.96	0.0506	0.5085
heights_phobia & video_scr_cmc & video_hp_cmc	0.1696	0.7501	0.23	0.8211	
heights_phobia & video_scr_cmc & resp_arousal_cmc	0.3066	0.6540	0.47	0.6392	
heights_phobia & video_hp_cmc & resp_arousal_cmc	-0.3282	0.1829	-1.79	0.0727	
video_scr_cmc & video_hp_cmc & resp_arousal_cmc	18.6106	9.7989	1.90	0.0575	9.0751
heights_phobia & video_scr_cmc & resp_valence_cmc	0.6375	0.4172	1.53	0.1266	

heights_phobia &	-0.0856	0.1270	-0.67	0.5003	
video_hp_cmc &					
resp_valence_cmc					
video_scr_cmc &	12.7287	5.7648	2.21	0.0272	2.5087
video_hp_cmc &					
resp_valence_cmc					
heights_phobia &					
resp_arousal_cmc &	0.1475	0.1127	1.31	0.1903	
resp_valence_cmc					
video_scr_cmc &					
resp_arousal_cmc &	-0.4491	4.5726	-0.10	0.9218	6.0251
resp_valence_cmc					
video_hp_cmc &					
resp_arousal_cmc &	3.8316	1.6401	2.34	0.0195	2.0238
resp_valence_cmc					
heights_phobia &					
video_scr_cmc &	-6.2221	4.2922	-1.45	0.1472	
video_hp_cmc &					
resp_arousal_cmc					
heights_phobia &					
video_scr_cmc &	-3.9741	2.4264	-1.64	0.1015	
video_hp_cmc &					
resp_valence_cmc					
heights_phobia &					
video_scr_cmc &	1.3406	1.9593	0.68	0.4938	
video_hp_cmc &					
resp_arousal_cmc &					
resp_valence_cmc					
heights_phobia &					
video_hp_cmc &	-0.9903	0.6975	-1.42	0.1557	
resp_arousal_cmc &					
resp_valence_cmc					
video_scr_cmc &					
video_hp_cmc &	49.9900	29.9565	1.67	0.0952	7.9913
resp_arousal_cmc &					
resp_valence_cmc					
heights_phobia &					
video_scr_cmc &					
video_hp_cmc &	-19.1634	12.0905	-1.59	0.1130	
resp_arousal_cmc &					
resp_valence_cmc					
Residual	0.1169				

```
1 mm14 = fit(LinearMixedModel, @formula(resp_fear ~ heights_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 244 Time: 0:00:00 (0.41 ms/it)
 objective: -1064.8565700333322

```
Minimizing 487      Time: 0:00:00 ( 0.62 ms/it)
objective: -1181.6591185338077
```

```
Minimizing 607      Time: 0:00:00 ( 0.68 ms/it)
objective: -1184.2949893660762
```

```
Minimizing 717      Time: 0:00:00 ( 0.72 ms/it)
objective: -1186.8524854822845
```

```
Minimizing 826      Time: 0:00:00 ( 0.75 ms/it)
objective: -1190.4581438380287
```

```
Minimizing 933      Time: 0:00:00 ( 0.77 ms/it)
objective: -1193.9565518959994
```

```
Minimizing 1040     Time: 0:00:00 ( 0.80 ms/it)
objective: -1197.4600729040446
```

```
mm15 =
```

	Est.	SE	z	p	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.5697	0.0733	7.77	<1e-14	0.1088
heights_phobia	0.0454	0.0136	3.35	0.0008	
video_scr_cmc	0.8534	0.2505	3.41	0.0007	0.4529
video_hp_cmc	0.0459	0.0602	0.76	0.4453	0.0464
resp_arousal_cmc	0.2279	0.0982	2.32	0.0204	0.2330
resp_valence_cmc	-0.5643	0.0850	-6.64	<1e-10	0.2453
video_scr_mean	-0.1475	0.2779	-0.53	0.5956	
video_hp_mean	-0.0625	0.0398	-1.57	0.1158	
resp_arousal_mean	0.2353	0.0540	4.36	<1e-04	
resp_valence_mean	-0.6330	0.0820	-7.72	<1e-13	
heights_phobia & video_scr_cmc	-0.1099	0.1017	-1.08	0.2797	
heights_phobia & video_hp_cmc	0.0046	0.0249	0.19	0.8527	
heights_phobia & resp_arousal_cmc	0.0310	0.0411	0.76	0.4500	
heights_phobia & resp_valence_cmc	-0.0151	0.0353	-0.43	0.6691	
Residual	0.1296				

```
1 mm15 = fit(LinearMixedModel, @formula(resp_fear ~ heights_phobia * video_scr_cmc + heights_phobia * video_hp_cmc + heights_phobia * resp_arousal_cmc + heights_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 19    Time: 0:00:00 ( 5.26 ms/it)
objective: -1013.017942427394
```

```
Minimizing 645    Time: 0:00:00 ( 0.24 ms/it)
```

mm16 =

	Est.	SE	z	p	σ_participant_num_str
(Intercept)	0.5851	0.0711	8.23	<1e-15	0.1274
heights_phobia	0.0396	0.0132	2.99	0.0028	
video_scr_cmc	0.7888	0.2622	3.01	0.0026	0.6303
video_hp_cmc	0.0246	0.0621	0.40	0.6917	0.0895
resp_arousal_cmc	0.2704	0.0783	3.45	0.0006	0.2349
resp_valence_cmc	-0.5479	0.0754	-7.27	<1e-12	0.2451
video_scr_mean	-0.0028	0.2668	-0.01	0.9915	
video_hp_mean	-0.0730	0.0365	-2.00	0.0456	
resp_arousal_mean	0.2168	0.0521	4.16	<1e-04	
resp_valence_mean	-0.6133	0.0788	-7.78	<1e-14	
heights_phobia & video_scr_cmc	-0.0944	0.1060	-0.89	0.3732	
heights_phobia & video_hp_cmc	0.0072	0.0264	0.27	0.7850	
heights_phobia & resp_arousal_cmc	0.0014	0.0332	0.04	0.9657	
heights_phobia & resp_valence_cmc	-0.0214	0.0313	-0.68	0.4935	
resp_arousal_cmc & resp_valence_cmc					0.5859
video_hp_cmc & resp_arousal_cmc & resp_valence_cmc					1.6447
video_scr_cmc & video_hp_cmc & resp_valence_cmc					4.4552
video_scr_cmc & video_hp_cmc					2.6447
video_scr_cmc & video_hp_cmc & resp_arousal_cmc					10.3441
video_scr_cmc & resp_valence_cmc					1.6713
video_scr_cmc & resp_arousal_cmc					2.5325
video_hp_cmc & resp_arousal_cmc					0.2098
video_scr_cmc & resp_arousal_cmc & resp_valence_cmc					6.2824
video_scr_cmc & video_hp_cmc & resp_arousal_cmc & resp_valence_cmc					8.5863
video_hp_cmc &					0.3925

```
resp_valence_cmc  
Residual          0.1172
```

```
1 mm16 = fit(LinearMixedModel, @formula(resp_fear ~ heights_phobia * video_scr_cmc  
+ heights_phobia * video_hp_cmc + heights_phobia * resp_arousal_cmc +  
heights_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 418      Time: 0:00:00 ( 0.44 ms/it)  
objective: -1131.3755734584138

Minimizing 645      Time: 0:00:00 ( 0.57 ms/it)  
objective: -1138.4191793738294

Minimizing 770      Time: 0:00:00 ( 0.61 ms/it)  
objective: -1141.9247872901099

Minimizing 889      Time: 0:00:00 ( 0.66 ms/it)  
objective: -1146.1046311407217

Minimizing 1002     Time: 0:00:00 ( 0.68 ms/it)  
objective: -1150.990419918598

Minimizing 1114     Time: 0:00:00 ( 0.70 ms/it)  
objective: -1155.0491159466733

Minimizing 1226     Time: 0:00:00 ( 0.72 ms/it)  
objective: -1157.6133945434274
```

Plot Test

	video	participant_num	resp_exp_fear	rt_exp_fear	resp_current_fear
1	"heights_low_3.mov"	107	0.048	2.503	0.045
2	"heights_low_2.m4v"	107	0.033	3.754	0.036
3	"heights_high_1.m4v"	107	0.032	2.52	0.024
4	"heights_high_3.m4v"	107	0.039	2.219	0.036
5	"heights_high_5.mov"	107	0.058	3.754	0.053
6	"heights_high_4.m4v"	107	0.282	4.355	0.17
7	"heights_low_6.m4v"	107	0.287	1.485	0.321
8	"heights_low_1.m4v"	107	0.05	2.319	0.044
9	"heights_low_4.m4v"	107	0.289	1.802	0.038
10	"heights_low_5.m4v"	107	0.059	1.368	0.056
: more					
1233	"heights_low_2.m4v"	232	0.264	4.722	0.081

```
1 dfcleancol2
```

► [0.675, 0.774, 0.931, 0.918, 0.807, 0.912, 0.518, 0.596, 0.797, 0.255, 0.814, 0.963, 0.

```
1 dfcleancol2.resp_arousal
```

	Est.	SE	z	p	σ_participant_num_str
(Intercept)	0.4594	0.0369	12.46	<1e-34	0.1806
video_scr_cmc	0.8985	0.1773	5.07	<1e-06	1.2116
video_scr_mean	1.0558	0.4809	2.20	0.0281	
Residual	0.2193				

```
1 mm3
```

```
randeff =
2×105 Matrix{Float64}:
0.175326 -0.065261 0.172386 0.109007 ... 0.103113 0.23012 -0.201264
0.703015 1.21637 0.72402 1.32789 ... 0.247021 0.0395284 0.99476
```

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

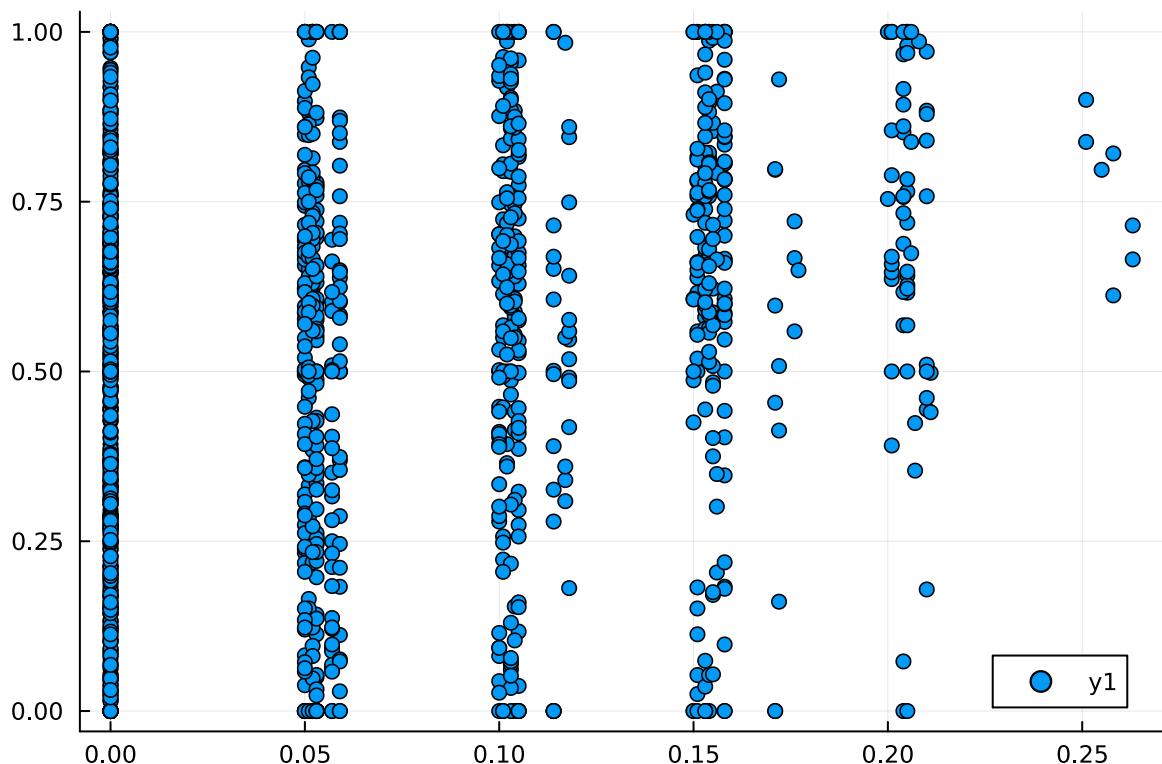
```
▶ [0.175326, -0.065261, 0.172386, 0.109007, 0.0524988, 0.150766, 0.0666498, 0.0416653, 0
```

```
1 raneff[1,:]
```

```
2×1 Matrix{Float64}:
```

```
0.17095474322780604  
0.8182084924876699
```

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



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

```
0.263
```

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

```
0.0
```

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

```
xax =  
10-element LinRange{Float64, Int64}:  
0.0, 0.0292222, 0.0584444, 0.0876667, 0.116889, 0.146111, 0.175333, 0.204556, 0.233778, 0.263  
1 xax = LinRange(minimum(dfcleancol2.video_scr),maximum(dfcleancol2.video_scr),10)
```

```
inter = 0.459362578400302
```

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

```
cmcbeta = 0.8984889479570923
```

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

```
meanbeta = 1.0557520457869927
```

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

```
yax =
```

```
10-element LinRange{Float64, Int64}:
0.459363, 0.51647, 0.573577, 0.630684, 0.687792, ..., 0.802006, 0.859113, 0.916221, 0.973328
```

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

```
► [1.7, 2.7, 2.25, 2.75, 3.2, 2.25, 1.95, 1.7, 2.85, 1.3, 1.3, 3.45, 4.05, 1.5, 1.5, 1.5, 1.8,
```

```
1 df2_subjlvl_hp.heights_phobia
```

```
0.9186874994315104
```

```
1 std(df2_subjlvl_hp.heights_phobia)
```

```
std_up = 3.2205922613362716
```

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

```
bm_std_up =
```

```
► BitVector: [false, false, false]
```

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

```
0.04398057803493485
```

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

```
std_dwn = 1.383217262473251
```

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

```
bm_std_dwn =
```

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

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

```
-0.044226358227131515
```

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

```
yax_up =
```

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

```
0.0439806, 0.0506916, 0.0574026, 0.0641136, ..., 0.0842465, 0.0909575, 0.0976685, 0.10438
```

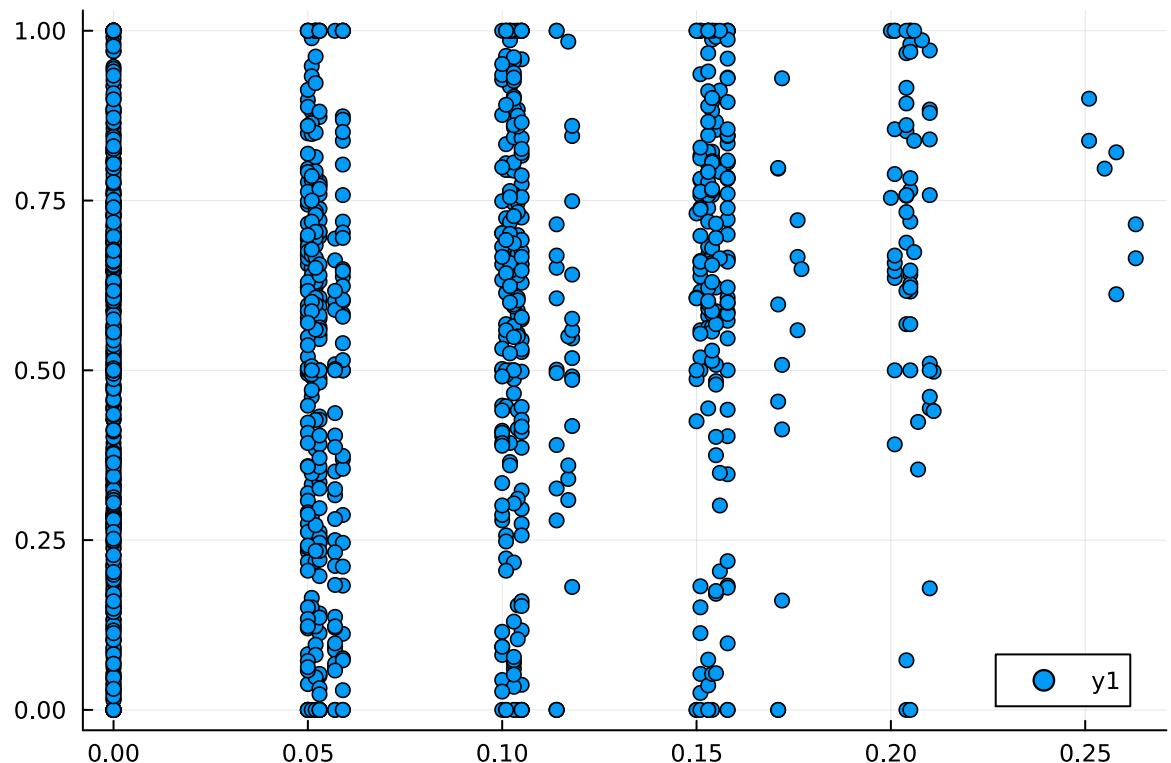
```
1 yax_up = mean(randeff[1,bm_std_up]) .+ (mean(randeff[2,bm_std_up]).* xax)
```

```
yax_dwn =  
► [0.0442264, 0.0572995, 0.0703726, 0.0834458, 0.0965189, 0.109592, 0.122665, 0.135738,  
1 yax_dwn = abs.(mean(randeff[1,bm_std_dwn]) .+ (mean(randeff[2,bm_std_dwn]).*  
xax))
```

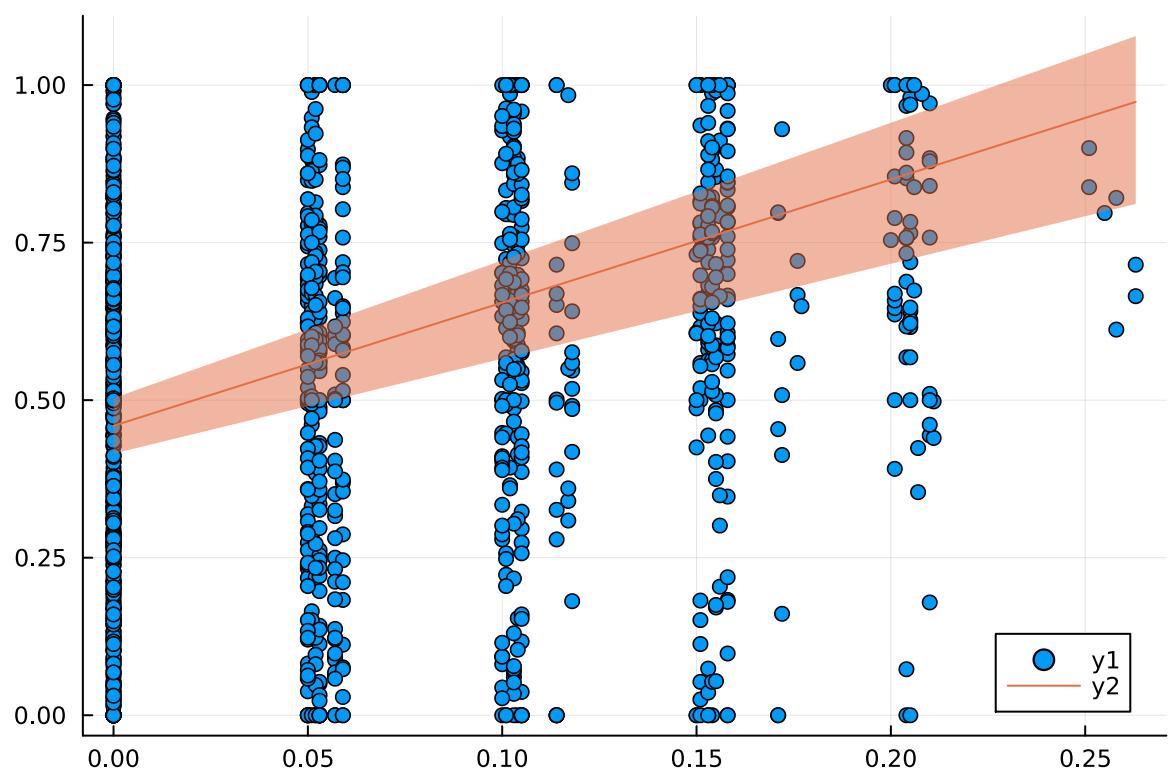
df =

	xax	yax	yax_up	yax_dwn
1	0.0	0.459363	0.0439806	0.0442264
2	0.0292222	0.51647	0.0506916	0.0572995
3	0.0584444	0.573577	0.0574026	0.0703726
4	0.0876667	0.630684	0.0641136	0.0834458
5	0.116889	0.687792	0.0708246	0.0965189
6	0.146111	0.744899	0.0775355	0.109592
7	0.175333	0.802006	0.0842465	0.122665
8	0.204556	0.859113	0.0909575	0.135738
9	0.233778	0.916221	0.0976685	0.148811
10	0.263	0.973328	0.10438	0.161885

```
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	$\sigma_{\text{participant_num_str}}$
(Intercept)	0.4594	0.0369	12.46	<1e-34	0.1806
video_scr_cmc	0.8985	0.1773	5.07	<1e-06	1.2116
video_scr_mean	1.0558	0.4809	2.20	0.0281	
Residual	0.2193				

1 mm3

▶ [2×105 Matrix{Float64}]:
0.175326 -0.065261 0.172386 0.109007 ... 0.103113 0.23012 -0.201264

1 ranef(mm3)

1 Enter cell code...

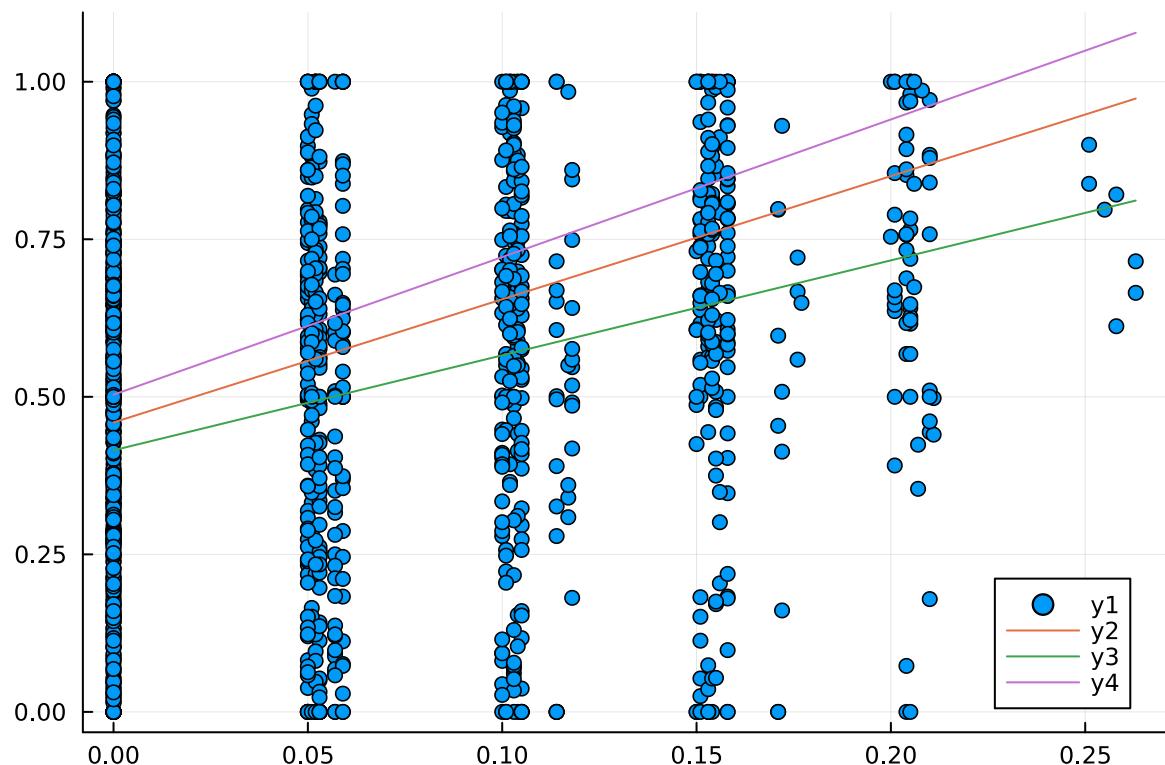
Plot Generic

plot_mm (generic function with 1 method)

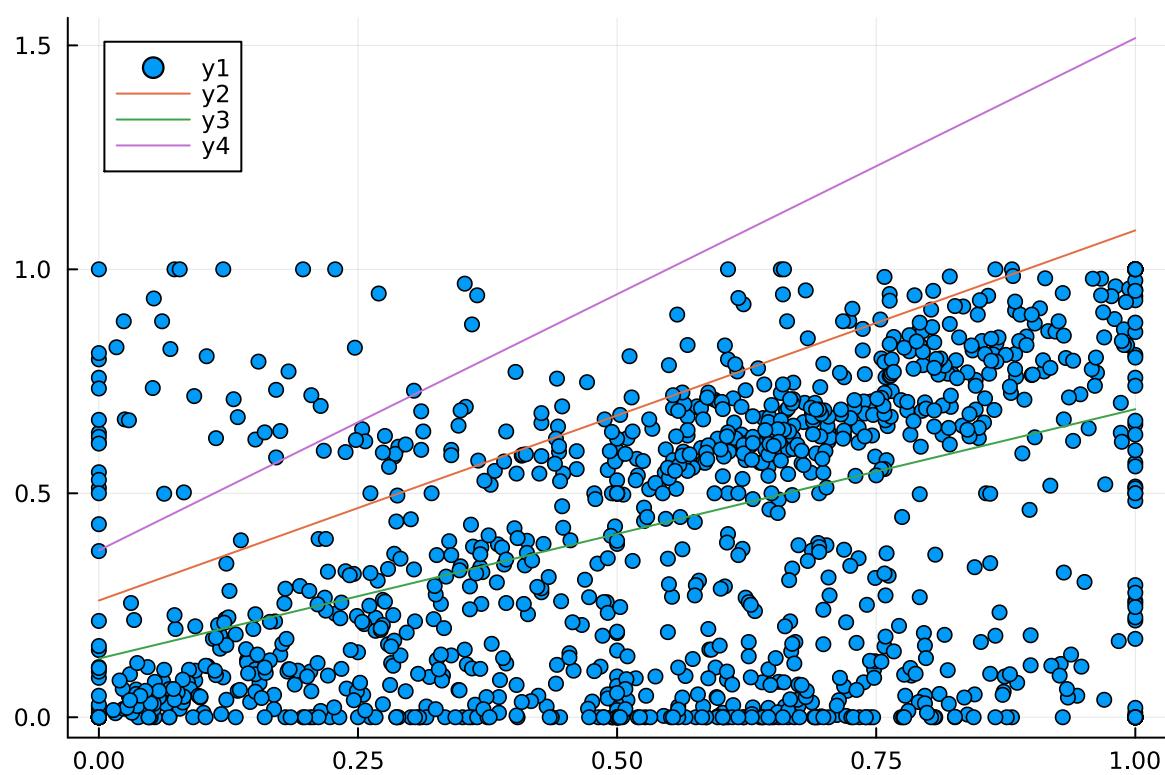
```

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.heights_phobia) +
9     std(df2_subjlvl_hp.heights_phobia)
10    bm_std_up = df2_subjlvl_hp.heights_phobia .> std_up
11    std_dwn = mean(df2_subjlvl_hp.heights_phobia) -
12    std(df2_subjlvl_hp.heights_phobia)
13    bm_std_dwn = df2_subjlvl_hp.heights_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)
end

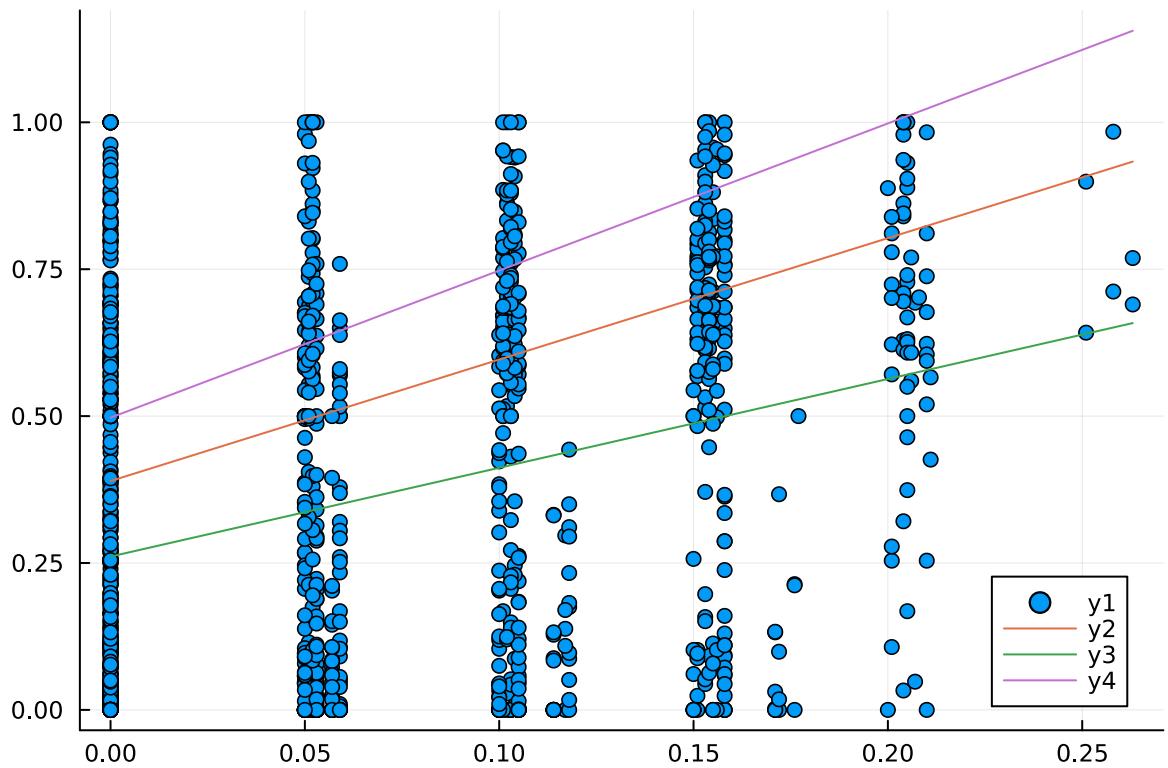
```



```
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 hparr = combine(gdf3, :heights_phobia => mean, renamecols=false).heights_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="Heights 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="Heights 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="Heights 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="Heights 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(m mz3))"
```

UndefVarError: mmz3 not defined

1. top-level scope @ Local: 1

```
1 mmz3coef_w = cor(ranef(m mz3)[1][2, :], hparr)
```

UndefVarError: mmz3 not defined

1. top-level scope @ Local: 1

```
1 mmz3coef_i = cor(ranef(m mz3)[1][1, :], hparr)
```

UndefVarError: mmz3 not defined

1. top-level scope @ Local: 1

```
1 scatter(ranef(m mz3)[1][1, :], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(m mz3coef_i)))", xlabel="SCR cluster mean centering MLM raneff  
Intercept", ylabel="Heights Phobia")
```

UndefVarError: mmz3 not defined

1. top-level scope @ Local: 1

```
1 scatter(ranef(m mz3)[1][2, :], hparr, label="participant", title="Src to arousal  
(cor=$(get_trunc(m mz3coef_w)))", xlabel="SCR cluster mean centering MLM raneff  
Weight", ylabel="Heights 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")
```