

Julia for R Lovers 2

July 14, 2021

1 Julia for R-Lovers

1.1 Demo: Sleepstudy LMM

```
[1]: using RCall;
      using MixedModels;
      using StatsBase, CSV, DataFrames;

      R"""
      library(tidyverse)
      library(lme4) #package for doing linear mixed effects models in R
      """
```

Warning: RCall.jl: Warning: replacing previous import 'vctrs::data_frame' by 'tibble::data_frame' when loading 'dplyr'

Attaching packages tidyverse 1.3.0

ggplot2 3.3.5	purrr 0.3.4
tibble 3.1.2	dplyr 1.0.0
tidyr 1.1.2	stringr 1.4.0
readr 1.3.1	forcats 0.5.0

Conflicts tidyverse_conflicts()

dplyr::filter() masks stats::filter()

dplyr::lag() masks stats::lag()

@ RCall /Users/kylamccconnell/.julia/packages/RCall/Qzssx/src/io.jl:160

Warning: RCall.jl: Loading required package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

expand, pack, unpack

@ RCall /Users/kylamccconnell/.julia/packages/RCall/Qzssx/src/io.jl:160

```
[1]: RObject{StrSxp}
      [1] "lme4"      "Matrix"    "forcats"   "stringr"   "dplyr"     "purrr"
```

```
[7] "readr"      "tidyr"      "tibble"     "ggplot2"    "tidyverse"  "stats"
[13] "graphics"   "grDevices"  "utils"      "datasets"   "methods"    "base"
```

1.1.1 Sleep study data

- Dataset included in lme4 in R and MixedModels in Julia
- 18 participants restricted to 3 hours of sleep every night for 9 nights
- DV: average reaction time speed
- <http://lme4.r-forge.r-project.org/slides/2011-01-11-Madison/2Longitudinal.pdf>

1.1.2 LMMs

- linear mixed effects models, add to linear regression the ability to account for random variance in repeated-measures designs (i.e., same participants or same items)
- technique well-used in psychology, cognitive science, linguistics, etc.
- lme4 models often fail to converge in R, requiring simplified model specification
- takes a long time even when it does converge

1.2 1) Load data in Julia

```
[2]: sleep = DataFrame(MixedModels.dataset(:sleepstudy));
names(sleep)
```

```
[2]: 3-element Array{String,1}:
 "subj"
 "days"
 "reaction"
```

```
[3]: first(sleep, 10)
```

```
[3]:
```

	subj	days	reaction
	String	Int8	Float64
1	S308	0	249.56
2	S308	1	258.705
3	S308	2	250.801
4	S308	3	321.44
5	S308	4	356.852
6	S308	5	414.69
7	S308	6	382.204
8	S308	7	290.149
9	S308	8	430.585
10	S308	9	466.353

```
[5]: summarystats(sleep.reaction)
```

```
[5]: Summary Stats:
      Length:      180
      Missing Count: 0
      Mean:        298.507892
      Minimum:     194.332200
      1st Quartile: 255.375825
      Median:      288.650800
      3rd Quartile: 336.752075
      Maximum:     466.353500
```

1.3 2) Wrangling in R

```
[6]: @rput sleep;
```

```
[7]: R"""
sleep %>%
  group_by(days) %>%
  summarize(mean(reaction))
"""
```

Warning: RCall.jl: `summarise()` ungrouping output (override with `.groups` argument)

@ RCall /Users/kylamccconnell/.julia/packages/RCall/Qzssx/src/io.jl:160

```
[7]: RObject{VecSxp}
# A tibble: 10 x 2
  days `mean(reaction)`
  <int>          <dbl>
1     0          257.
2     1          264.
3     2          265.
4     3          283.
5     4          289.
6     5          309.
7     6          312.
8     7          319.
9     8          337.
10    9          351.
```

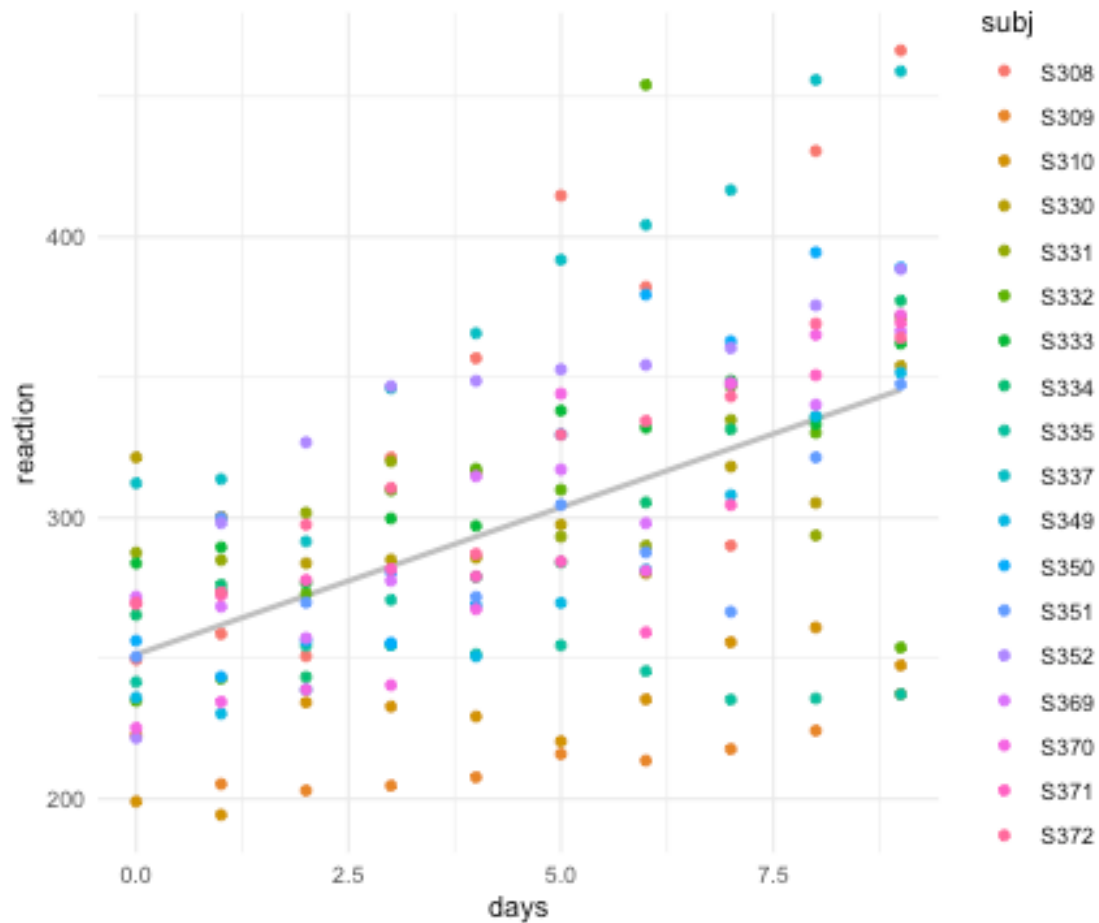
```
[8]: R"""
sleep %>%
  group_by(subj) %>%
  summarize(mean(reaction))
"""
```

```
Warning: RCall.jl: `summarise()` ungrouping output (override with `.groups`  
argument)
```

```
@ RCall /Users/kylamccconnell/.julia/packages/RCall/Qzssx/src/io.jl:160
```

```
[8]: RObject{VecSxp}  
# A tibble: 18 x 2  
  subj   `mean(reaction)`  
  <chr>         <dbl>  
1 S308          342.  
2 S309          215.  
3 S310          231.  
4 S330          303.  
5 S331          309.  
6 S332          307.  
7 S333          316.  
8 S334          295.  
9 S335          250.  
10 S337          376.  
11 S349          276.  
12 S350          314.  
13 S351          290.  
14 S352          337.  
15 S369          306.  
16 S370          292.  
17 S371          295.  
18 S372          318.
```

```
[10]: R""  
ggplot(sleep, aes(x= days, y = reaction)) +  
  geom_smooth(method = "lm", color = "grey", se = F) +  
  geom_point(aes(color = subj), position = "dodge") +  
  theme_minimal()  
""
```



[10]: RObject{VecSxp}

Warning: RCall.jl: `geom_smooth()` using formula 'y ~ x'
 Warning: Width not defined. Set with `position_dodge(width = ?)`
 @ RCall /Users/kylamcconnell/.julia/packages/RCall/Qzssx/src/io.jl:160

```
[11]: R"""
ggplot(sleep, aes(x= days, y = reaction)) +
  geom_smooth(method = "lm", color = "grey", se = F) +
  geom_point(aes(color = as.factor(days)), position = "dodge", show.legend =
↪FALSE) +
  facet_wrap(~subj) +
  theme_minimal()
"""
```



```
[11]: RObject{VecSxp}
```

```
Warning: RCall.jl: `geom_smooth()` using formula 'y ~ x'
Warning: Width not defined. Set with `position_dodge(width = ?)`
@ RCall /Users/kylamcconnell/.julia/packages/RCall/Qzssx/src/io.jl:160
```

1.4 3) Model in Julia

```
[12]: @rget sleep; #don't have to do this in this example, but would have to if you
      ↪make changes to the df in R
```

LMM formula (similar to R) Regression syntax - DV ~ predictors Random effect term: - accounts for difference by subj - random intercepts (y-axis location) - random slope - (1 + predictor | subj)

- In this case MixedModel syntax is similar to R:
 - `lmer(reaction ~ days + (1 + days | subj))`

```
[14]: formula_sleep = @formula (reaction ~ days + (1 + days | subj));
```

```
[15]: sleep_model = fit(MixedModel, formula_sleep, sleep);
```

```
[16]: show(sleep_model)
```

Linear mixed model fit by maximum likelihood

reaction ~ 1 + days + (1 + days | subj)

	logLik	-2 logLik	AIC	AICc	BIC
	-875.9697	1751.9393	1763.9393	1764.4249	1783.0971

Variance components:

	Column	Variance	Std.Dev.	Corr.
subj	(Intercept)	565.51069	23.78047	
	days	32.68212	5.71683	+0.08

Residual 654.94145 25.59182

Number of obs: 180; levels of grouping factors: 18

Fixed-effects parameters:

	Coef.	Std. Error	z	Pr(> z)
(Intercept)	251.405	6.63226	37.91	<1e-99
days	10.4673	1.50224	6.97	<1e-11

1.4.1 Example

```
formula_maximal = @formula (DV ~ f_1 * f_2 * f_3 * f_4 + c_1 + c_2 + c_3 + c_4 +  
c_5 + (1 + f_1 + c_1 + c_2 + c_3 + c_4 | id) + (1 + c_1 + f_2 * f_3 | item_1) +  
(1 + c_1 + f_2 * f_3 | item_2));
```

1.4.2 Coding categorical predictors

```
cntrsts = merge( Dict(:cond => EffectsCoding(base=~`cond_A`), :education =>  
HelmertCoding(levels=[~`High school`, ~`Undergraduate`, ~`Grad school`]), :id  
=> Grouping(), :item => Grouping()) );
```

```
sleep_model = fit(MixedModel, formula_sleep, sleep, contrasts = cntrsts);
```

1.5 Visualize model output in R

```
[17]: using JellyMe4 #companion to lme4 / MixedModels and RCall
```

```
sleep_model_R = (sleep_model, sleep)
```

```
@rput sleep_model_R
```

```
[17]: (Linear mixed model fit by maximum likelihood
      reaction ~ 1 + days + (1 + days | subj)
      logLik   -2 logLik       AIC       AICc       BIC
      -875.9697 1751.9393 1763.9393 1764.4249 1783.0971
```

Variance components:

	Column	Variance	Std.Dev.	Corr.
subj	(Intercept)	565.51069	23.78047	
	days	32.68212	5.71683	+0.08
Residual		654.94145	25.59182	

Number of obs: 180; levels of grouping factors: 18

Fixed-effects parameters:

	Coef.	Std. Error	z	Pr(> z)
(Intercept)	251.405	6.63226	37.91	<1e-99
days	10.4673	1.50224	6.97	<1e-11

, 180x3 DataFrame

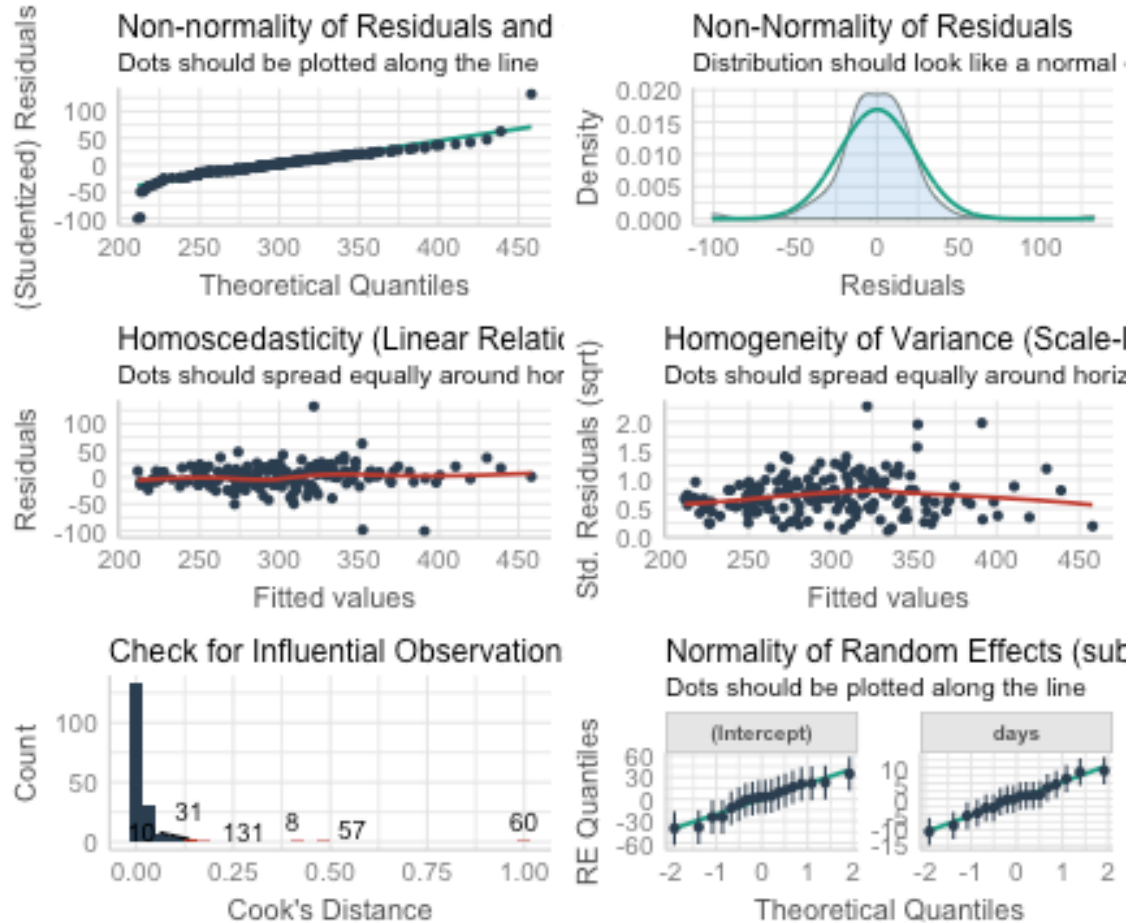
Row	subj	days	reaction
	String	Int64	Float64
1	S308	0	249.56
2	S308	1	258.705
3	S308	2	250.801
4	S308	3	321.44
5	S308	4	356.852
6	S308	5	414.69
7	S308	6	382.204
8	S308	7	290.149
9	S308	8	430.585
10	S308	9	466.353
170	S371	9	369.469
171	S372	0	269.412
172	S372	1	273.474
173	S372	2	297.597
174	S372	3	310.632
175	S372	4	287.173
176	S372	5	329.608
177	S372	6	334.482
178	S372	7	343.22
179	S372	8	369.142
180	S372	9	364.124)

```
[18]: R"""
      library(performance)
```



```
check_model(sleep_model_R)
""
```

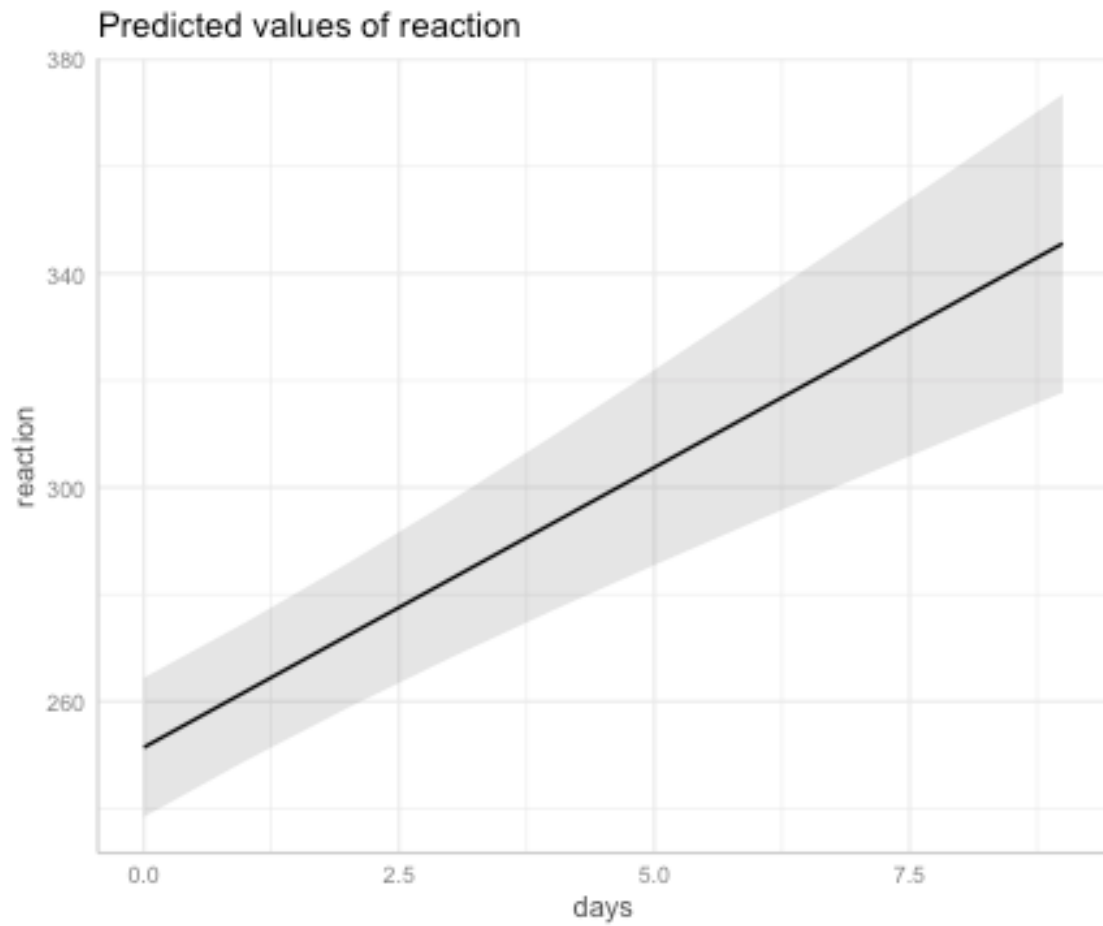
Not enough model terms in the conditional part of the model to check for multicollinearity.



```
[18]: RObject{VecSxp}
```

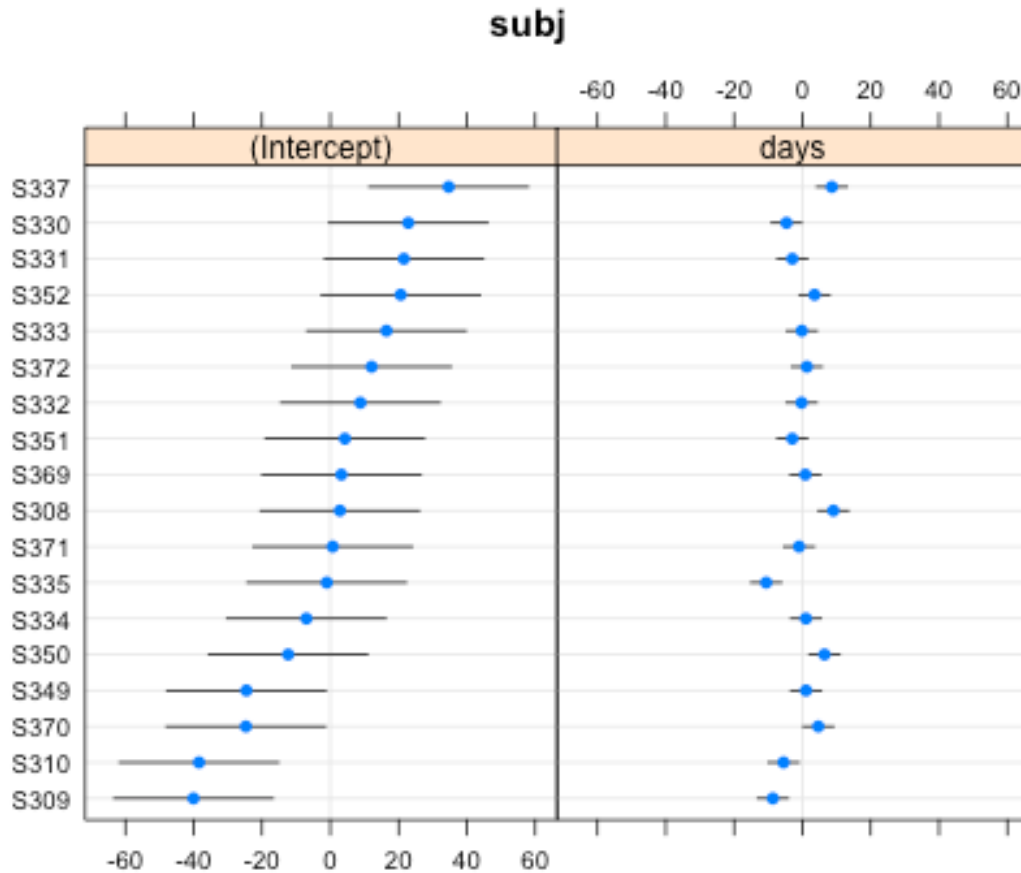
```
Warning: RCall.jl: Warning: `guides(<scale> = FALSE)` is deprecated. Please
use `guides(<scale> = "none")` instead.
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
Warning: Removed 174 rows containing missing values (geom_text_repel).
`geom_smooth()` using formula 'y ~ x'
@ RCall /Users/kylamccconnell/.julia/packages/RCall/Qzssx/src/io.jl:160
```

```
[19]: R"""  
library(ggeffects)  
plot(ggpredict(sleep_model_R, terms = "days"))  
"""
```



```
[19]: RObject{VecSxp}
```

```
[20]: R"""  
library(lattice)  
dotplot(ranef(sleep_model_R))  
"""
```



```
[20]: RObject{VecSxp}
      $subj
```

1.6 Summary

- With the R commands from RCall (`R"```, `@rput`, `@rget`) you can use R for visualization and wrangling but let Julia do the "heavy lifting" of modeling
- My example uses LMMs (my use case) but you could substitute that step with any modeling methodology

Things to look out for: - missing values may be treated differently - easy solution: remove NAs in R in advance - changes in packages, especially "younger" ones - may have to be creative with package management - may be less on Stack Overflow