

Multilevel Model Estimation Using the Lueven Clinical Data Set

Ginette Lafit

2023-06-06

Table of contents

Settings things up	1
Description	3
Data exploration	4
Data structure	4
Descriptive statistics and visualizations	6
Example 1	19
Example 2	24
Example 3	28
Session information	33
References	34

Settings things up

Before we proceed, we need to ensure we have several packages installed and loaded into our R session. For the scripts below, we will use the following packages:

- `tidyverse`
- `data.table`
- `psych`
- `viridis`
- `devtools`
- `nlme`

Which we can install in one go as follows:

```
# Prepare the package list.
packages = c(
  "tidyverse", "data.table", "psych",
  "viridis", "devtools", "nlme"
)

# Install packages.
install.packages(packages)
```

Tip

You may consider first checking if the packages are installed before actually installing them. Nevertheless, the code above will not reinstall packages that are already installed and up-to-date.

Now that we have all packages installed, we continue by loading them.

```
# Handy collection of packages for data manipulation and plotting.
library(tidyverse)

# To create lagged outcome.
library(data.table)

# To compute descriptive statistics.
library(psych)

# Color scales adapted for colorblindness.
library(viridis)

# To estimate mixed-effects models.
library(nlme)
```

Additionally, we may also need to install and load the PowerLAPIM package from GitHub:

```
# Install
devtools::install_github("ginettelafit/PowerLAPIM", force = TRUE)
```

To complete the setup, we also need to set the seed for reproducibility.

```
# Set a seed for reproducibility.
set.seed(123)
```

Description

In this tutorial, we use data from Heininga et al. (2019). In this study, the authors applied the ESM methodology to study emotion dynamics in people with *Major Depressive Disorder* (MDD). The study consist of an ESM testing period of seven days in which participants had to fill out questions about mood and social context on their daily lives ten times a day (i.e., 70 measurement occasions). The data set contains 38 participants diagnosed with MDD and 40 control subjects. Participants filled out the ESM questionnaires in a stratified random interval scheme between 9:30 AM and 9:30 PM.

The data set contains the following variables:

- PID that denotes the individual identification number
- day is a variable that ranges from 1 to 7 and identifies the day of ESM testing
- daybeep is a variable that ranges from 1 to 10 and identifies the number of the prompt or beep within a day
- PA is the *positive affect* computed as the mean of items:
 - *How happy do you feel at the moment?*
 - *How relaxed do you feel at the moment?*
 - *How euphoric do you feel at the moment?*
- NA. is the *negative affect* computed as the mean of items:
 - *How depressed do you feel at the moment?*
 - *How stressed do you feel at the moment?*
 - *How anxious do you feel at the moment?*
 - *How angry do you feel at the moment?*
 - *How restless do you feel at the moment?*
- anhedonia corresponds to the ESM item:
 - *To what degree do you find it difficult to experience pleasure in activities at the moment?*
- MDD is a dummy variable equal to one when the individual has been diagnosed with MDD and 0 otherwise
- QIDS denotes the sum of the items of the *Quick Inventory of Depressive Symptomatology* (QIDS; Rush et al., 2003). QIDS was measured before the ESM testing period.

First, we are going to load the data set:

```
# Load the data set.  
load(file = "assets/data/clinical-dataset.RData")
```

Tip

Make sure you load the data from the location where you downloaded it. If your analysis script (i.e., the .R file) and the dataset are in the same location, then you can simply load the data as follows:

```
load(file = "clinical-dataset.RData")
```

Data exploration

In this section we will explore briefly the variables in the data set.

Data structure

Now, that we have the data ready, we can start by exploring it to get a better understanding of the variable measured.

```
# Find the dimensions.  
dim(data)
```

```
[1] 5460    8
```

```
# Find the structure.  
str(data)
```

```
'data.frame':  5460 obs. of  8 variables:  
 $ PID      : num  101 101 101 101 101 101 101 101 101 101 ...  
 $ day       : num   1  1  1  1  1  1  1  1  1  1 ...  
 $ daybeep   : num   1  2  3  4  5  6  7  8  9 10 ...  
 $ PA        : num   NA 27.3 49.7 43 43 ...  
 $ NA.       : num   NA 30.4 23.8 24.2 32.8 19.6 18.4 21.2 23 21.8 ...  
 $ anhedonia : num   NA 26 25 25 50 21 42 30 22 30 ...  
 $ MDD       : num   1  1  1  1  1  1  1  1  1  1 ...  
 $ QIDS      : num  12 12 12 12 12 12 12 12 12 12 ...
```

```
# See the first 6 rows.  
head(data)
```

	PID	day	daybeep	PA	NA.	anhedonia	MDD	QIDS
1	101	1	1	NA	NA	NA	1	12
2	101	1	2	27.33333	30.4	26	1	12
3	101	1	3	49.66667	23.8	25	1	12
4	101	1	4	43.00000	24.2	25	1	12
5	101	1	5	43.00000	32.8	50	1	12
6	101	1	6	18.00000	19.6	21	1	12

```
# See the last 6 rows.
```

```
tail(data)
```

	PID	day	daybeep	PA	NA.	anhedonia	MDD	QIDS
5455	645	7	5	70.66667	9.4	10	0	4
5456	645	7	6	73.66667	11.0	20	0	4
5457	645	7	7	64.33333	10.8	18	0	4
5458	645	7	8	69.66667	11.2	10	0	4
5459	645	7	9	73.33333	13.0	18	0	4
5460	645	7	10	65.66667	15.2	15	0	4

```
# Find the column names.
```

```
names(data)
```

```
[1] "PID"      "day"      "daybeep"  "PA"      "NA."      "anhedonia"
[7] "MDD"      "QIDS"
```

```
# Summary of the data.
```

```
summary(data)
```

PID		day		daybeep		PA		NA.	
Min.	:101.0	Min.	:1	Min.	: 1.0	Min.	: 0.00	Min.	: 0.00
1st Qu.	:131.0	1st Qu.	:2	1st Qu.	: 3.0	1st Qu.	:23.00	1st Qu.	: 6.60
Median	:601.5	Median	:4	Median	: 5.5	Median	:36.33	Median	: 18.80
Mean	:390.2	Mean	:4	Mean	: 5.5	Mean	:37.07	Mean	: 25.61
3rd Qu.	:624.0	3rd Qu.	:6	3rd Qu.	: 8.0	3rd Qu.	:50.00	3rd Qu.	: 40.30
Max.	:645.0	Max.	:7	Max.	:10.0	Max.	:93.67	Max.	:100.00
						NA's	:629	NA's	:629
anhedonia		MDD		QIDS					
Min.	: 0.00	Min.	:0.0000	Min.	: 0.000				
1st Qu.	: 7.00	1st Qu.	:0.0000	1st Qu.	: 3.000				

```

Median : 28.00   Median :0.0000   Median : 8.000
Mean   : 33.34   Mean    :0.4872   Mean    : 9.359
3rd Qu.: 56.00   3rd Qu.:1.0000   3rd Qu.:16.000
Max.   :100.00   Max.    :1.0000   Max.    :24.000
NA's   :629

```

```

# Number of participants.
length(unique(data$PID))

```

```
[1] 78
```

```

# Create variable to store the number of observations per person.
data$obs = rep(0, nrow(data))

# Count the number of observation per person.
for (i in unique(data$PID)) {
  data$obs[which(data$PID == i)] <- 1:length(which(data$PID == i))
}

# Show the number of observations per person.
table(data$obs)

```

```

 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70
78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78

```

Descriptive statistics and visualizations

We first compute descriptive statistics including number of participant, number of observations per day, and compliance.

```

# Get number of participants.
length(unique(data$PID))

```

```
[1] 78
```

```
# Obtain number of participants diagnosed with `MDD`.
length(unique(data$PID[data$MDD == 1]))
```

```
[1] 38
```

```
# Obtain number of participants in the control group.
length(unique(data$PID[data$MDD == 0]))
```

```
[1] 40
```

```
# Get the number of assessment per day.
table(data$PID)
```

```
101 102 103 105 108 109 110 111 112 114 115 117 119 120 121 124 125 128 130 131
 70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70
132 135 136 138 139 140 141 143 144 202 204 205 206 207 208 210 211 306 601 602
 70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70
603 604 606 607 608 609 610 611 612 613 614 615 616 618 619 621 622 623 624 625
 70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70
626 627 628 629 630 631 632 634 635 636 637 638 640 641 642 643 644 645
 70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70
```

```
# Get the number of assessment per day for each participant.
beeps.person <- lapply(
  data$PID, function(i) {
    table(data$day[which(data$PID == i)])
  }
)
```

```
# Show results for some of the participants.
beeps.person[1:6]
```

```
[[1]]
```

```
 1  2  3  4  5  6  7
10 10 10 10 10 10 10
```

```
[[2]]
```

```

 1  2  3  4  5  6  7
10 10 10 10 10 10 10

```

```
[[3]]
```

```

 1  2  3  4  5  6  7
10 10 10 10 10 10 10

```

```
[[4]]
```

```

 1  2  3  4  5  6  7
10 10 10 10 10 10 10

```

```
[[5]]
```

```

 1  2  3  4  5  6  7
10 10 10 10 10 10 10

```

```
[[6]]
```

```

 1  2  3  4  5  6  7
10 10 10 10 10 10 10

```

```

# Compute a binary variable indicating if a participant answered a beep. We take
# the ESM item PA as reference because in this ESM design participants were not
# allowed to skip items.

```

```
data$Compliance <- ifelse(is.na(data$PA) == FALSE, 1, 0)
```

```

# Mean, median of the compliance across all participants.
describe(data$Compliance)

```

```

vars      n mean   sd median trimmed mad min max range  skew kurtosis se
X1      1 5460 0.88 0.32      1    0.98  0  0  1      1 -2.41    3.81  0

```

```
# Compliance per participant.
```

```

data.compliance.person <- aggregate(
  data$Compliance,
  by = list(data$PID),
  mean,
  na.rm = TRUE
)

```



```
)

# See the first 6 rows.
head(data.compliance.person)
```

```
Group.1      x
1      101 0.9142857
2      102 0.8857143
3      103 0.9571429
4      105 0.9714286
5      108 0.6000000
6      109 0.9857143
```

```
# See the last 6 rows.
tail(data.compliance.person)
```

```
Group.1      x
73      640 0.7714286
74      641 0.8571429
75      642 0.9428571
76      643 0.9857143
77      644 0.7142857
78      645 0.9571429
```

```
# Obtain descriptive statistics of person's average compliance.
describe(data.compliance.person$x)
```

```
vars  n mean  sd median trimmed  mad  min max range  skew kurtosis  se
X1    1 78 0.88 0.1  0.92    0.9 0.07 0.54   1  0.46 -1.29    1.28 0.01
```

Next, we obtain descriptive statistics of the distribution of the person-level or time-invariant variables.

```
# We create a variable including the
# diagnosis (i.e. 1 = `MDD` and 0 = control group),
# and depression (`QIDS`) for each participant.
dt.person <- aggregate(
  cbind(data$MDD, data$QIDS),
```

```

    by = list(data$PID),
    mean,
    na.rm = TRUE
)

# Add column names.
colnames(dt.person) <- c("Group.1", "MDD", "QIDS")

# See the first 6 rows.
head(dt.person)

```

	Group.1	MDD	QIDS
1	101	1	12
2	102	1	10
3	103	1	18
4	105	1	16
5	108	1	5
6	109	1	14

```

# See the last 6 rows.
tail(dt.person)

```

	Group.1	MDD	QIDS
73	640	0	1
74	641	0	4
75	642	0	0
76	643	0	6
77	644	0	10
78	645	0	4

```

# Descriptive statistics for time-invariant variable `QIDS`.
describe(dt.person$QIDS)

```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	78	9.36	7.35	8	8.92	8.9	0	24	24	0.43	-1.25	0.83

```

# Descriptive statistics for time-invariant variable `QIDS` for `MDD = 1`.
describe(dt.person$QIDS[dt.person$MDD == 1])

```

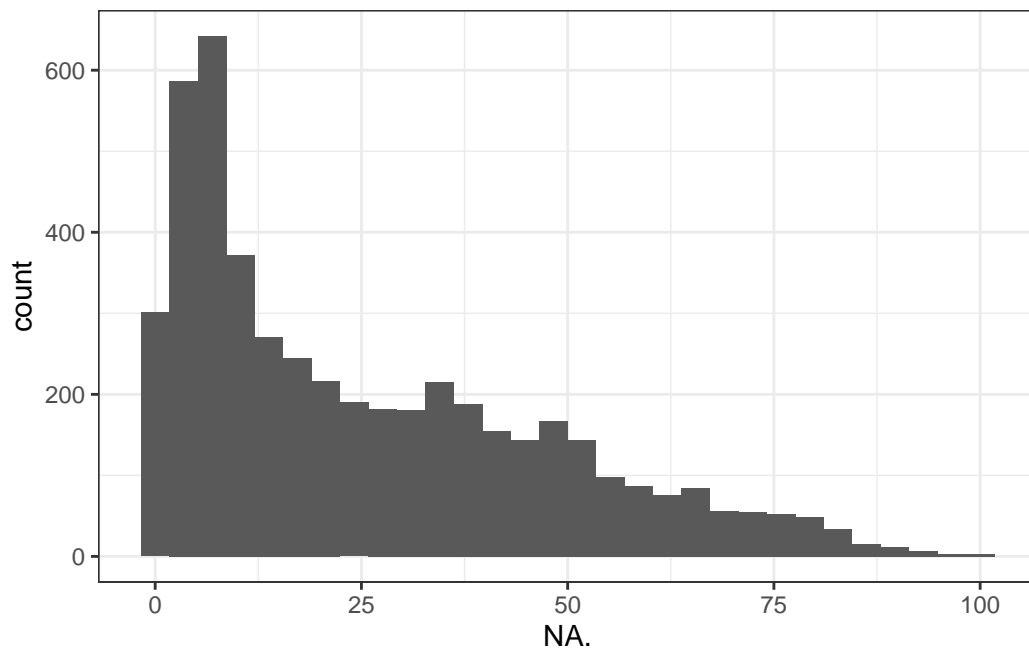
```
vars  n  mean sd median trimmed  mad min max range skew kurtosis  se
X1    1 38 15.71 5     16   15.88 5.93  5 24   19 -0.3    -0.82 0.81
```

```
# Descriptive statistics for time-invariant variable `QIDS` for `MDD = 0`.
describe(dt.person$QIDS[dt.person$MDD == 0])
```

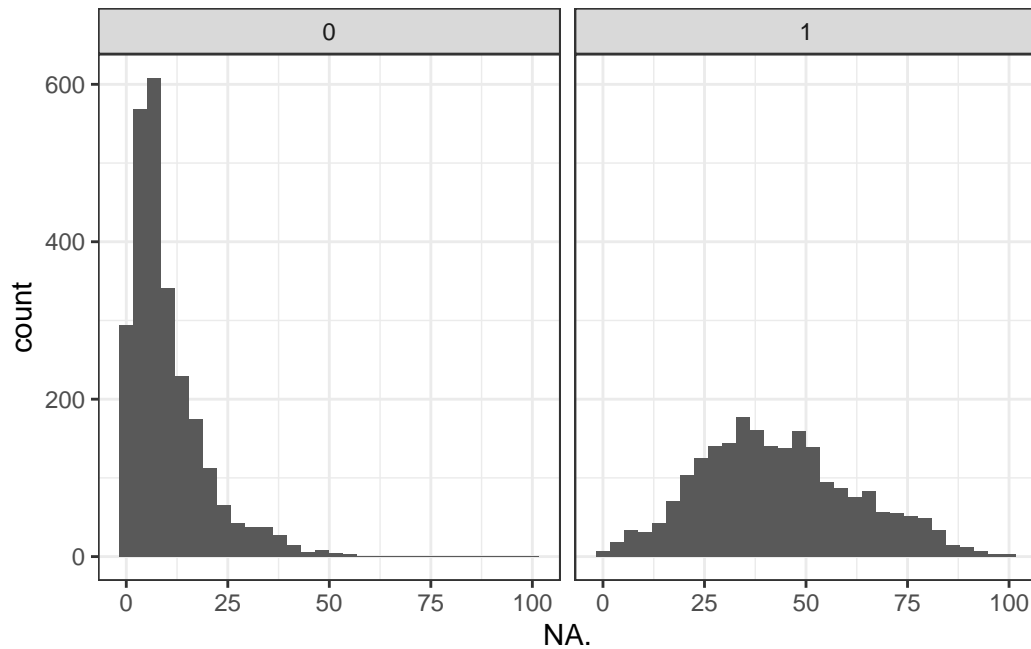
```
vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X1    1 40 3.33 2.53     3    3.03 1.48  0 10   10 0.94    0.54 0.4
```

We now focus the time-varying variables, we obtain visualization and descriptive statistics for the time-varying variable negative affect (NA).

```
# Histogram for the time-varying variable negative affect (i.e. `NA.`).
ggplot(data, aes(NA.)) +
  geom_histogram(
    bins = 30
  ) +
  scale_fill_viridis() +
  theme_bw()
```



```
# Histogram for the time-varying variable `NA.` by `MDD`.
ggplot(data, aes(NA.)) +
  geom_histogram(
    bins = 30
  ) +
  facet_wrap(
    . ~ MDD
  ) +
  scale_fill_viridis() +
  theme_bw()
```



```
# Descriptive statistics for `NA.`.
describe(data$NA.)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	4831	25.61	22.19	18.8	22.95	21.05	0	100	100	0.86	-0.17	0.32

```
# Descriptive statistics for `NA.` in the `MDD` group.
describe(data$NA.[data$MDD == 1])
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
--	------	---	------	----	--------	---------	-----	-----	-----	-------	------	----------	----

```
X1      1 2255 43.28 19.22   41.2   42.57 19.57   0 100   100 0.31   -0.39 0.4
```

```
# Descriptive statistics for `NA.` in the control group.
describe(data$NA.[data$MDD == 0])
```

```
vars      n mean   sd median trimmed  mad min max range skew kurtosis   se
X1       1 2576 10.15 9.34    7.2    8.68 6.23   0 66   66 1.75    3.77 0.18
```

```
# Distribution of happy per participant.
data.table.dt <- setDT(na.omit(data))
data.table.dt[, as.list(summary(NA., na.omit = TRUE)), by = PID]
```

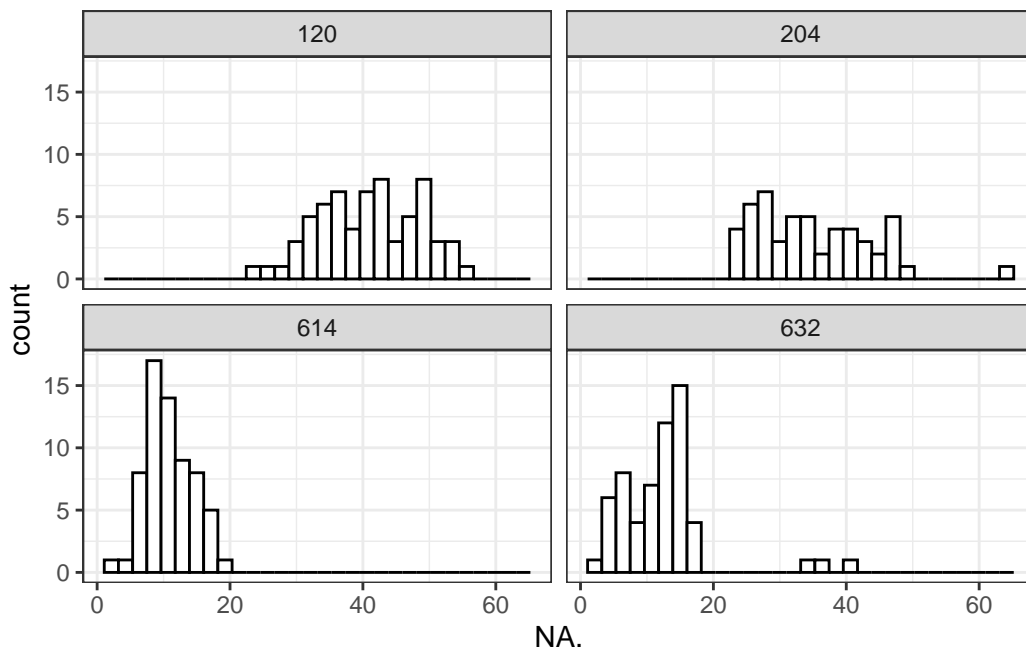
	PID	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1:	101	14.4	21.40	25.9	29.621875	36.25	65.6
2:	102	14.4	32.40	38.2	39.090323	46.25	64.2
3:	103	25.4	57.50	66.6	65.528358	75.10	93.2
4:	105	25.2	42.80	50.7	50.235294	57.40	77.8
5:	108	9.0	13.40	17.1	17.647619	21.35	37.6
6:	109	11.6	25.80	32.2	33.956522	43.20	60.8
7:	110	7.8	20.60	29.4	31.664615	40.60	68.6
8:	111	13.2	24.40	26.3	26.767647	30.05	45.6
9:	112	47.4	62.50	66.2	68.817910	73.90	92.2
10:	114	5.6	27.85	35.3	36.342424	44.45	73.8
11:	115	18.0	42.40	56.2	57.168627	77.80	84.2
12:	117	22.2	27.80	35.4	34.553125	40.05	51.4
13:	119	12.8	26.80	38.6	37.927869	46.00	91.0
14:	120	24.4	35.30	40.4	40.857576	46.60	54.6
15:	121	31.6	44.85	49.2	50.421429	56.65	73.0
16:	124	15.0	27.60	33.4	34.266667	39.80	69.0
17:	125	13.8	19.00	23.2	23.567347	27.20	39.4
18:	128	8.0	33.70	49.1	45.072414	58.85	69.8
19:	130	10.8	32.40	52.8	49.652830	68.40	83.6
20:	131	19.0	38.80	53.6	49.932308	60.00	75.0
21:	132	15.2	32.60	43.4	43.393333	52.10	81.4
22:	135	18.0	47.55	57.4	55.561765	64.60	91.2
23:	136	0.0	4.85	15.5	25.842424	35.85	100.0
24:	138	25.0	40.55	47.8	47.955882	53.40	76.6
25:	139	22.0	35.20	42.0	41.487719	48.20	57.0
26:	140	38.8	47.50	52.7	52.700000	57.35	78.0
27:	141	18.4	30.35	37.3	36.042105	40.50	50.8

28:	143	59.4	72.60	75.1	74.590909	77.55	82.4
29:	144	45.2	50.75	53.1	54.143333	55.25	74.2
30:	202	5.0	27.30	34.4	35.142424	40.25	66.6
31:	204	22.6	28.35	34.3	35.153846	40.95	65.2
32:	205	15.8	28.35	34.4	34.539286	42.25	52.6
33:	206	3.6	7.05	9.4	13.531034	18.70	50.6
34:	207	27.2	78.40	82.4	81.207547	88.60	100.0
35:	208	25.0	56.00	64.6	63.627692	71.80	96.4
36:	210	15.4	32.75	40.5	44.350000	53.40	84.6
37:	211	27.2	34.40	38.8	39.609836	43.40	56.6
38:	306	5.2	21.00	30.2	30.680702	39.80	67.2
39:	601	0.0	2.40	5.1	9.190909	10.95	42.6
40:	602	2.0	4.40	6.0	6.453731	7.80	15.4
41:	603	0.0	6.15	14.1	16.556250	24.00	56.6
42:	604	3.2	9.80	12.8	13.620000	16.15	39.2
43:	606	0.0	6.80	9.1	9.225000	12.00	18.6
44:	607	0.0	0.40	10.2	12.200000	20.30	63.0
45:	608	0.0	1.00	8.0	10.140299	14.20	52.6
46:	609	4.8	24.00	29.2	29.714286	37.00	48.2
47:	610	5.6	16.95	24.6	26.059375	33.05	66.0
48:	611	1.6	4.00	5.6	6.552941	7.10	26.0
49:	612	2.2	3.40	5.2	5.857143	6.90	17.6
50:	613	0.0	0.00	1.6	3.609231	3.80	36.2
51:	614	3.2	8.80	10.4	10.865625	13.20	18.8
52:	615	0.0	0.00	0.0	1.368571	0.20	37.0
53:	616	0.0	0.00	2.7	5.758824	9.00	40.4
54:	618	0.0	2.20	3.4	4.174194	4.60	17.4
55:	619	1.0	3.50	5.0	5.114286	6.55	14.6
56:	621	9.8	16.10	20.2	20.682353	24.10	32.8
57:	622	7.6	13.55	16.2	16.515625	19.35	29.0
58:	623	2.0	5.40	6.6	7.442857	8.60	23.4
59:	624	2.4	6.40	9.5	10.533333	14.75	21.4
60:	625	0.8	5.40	7.8	14.073846	17.40	47.0
61:	626	0.8	2.20	3.6	6.636066	5.60	48.8
62:	627	0.0	1.65	4.0	5.719355	8.10	27.0
63:	628	0.8	3.40	5.2	6.206154	6.60	25.2
64:	629	2.6	8.80	12.6	13.993846	17.20	41.0
65:	630	1.2	4.60	6.0	10.284848	10.90	37.6
66:	631	1.6	3.80	6.0	6.057143	7.40	16.6
67:	632	3.2	7.55	13.0	12.473333	14.45	40.0
68:	634	0.0	1.30	3.6	4.208955	5.80	22.0
69:	635	1.8	4.35	5.3	7.062500	6.20	38.0
70:	636	0.0	4.05	6.0	6.120000	8.15	12.4

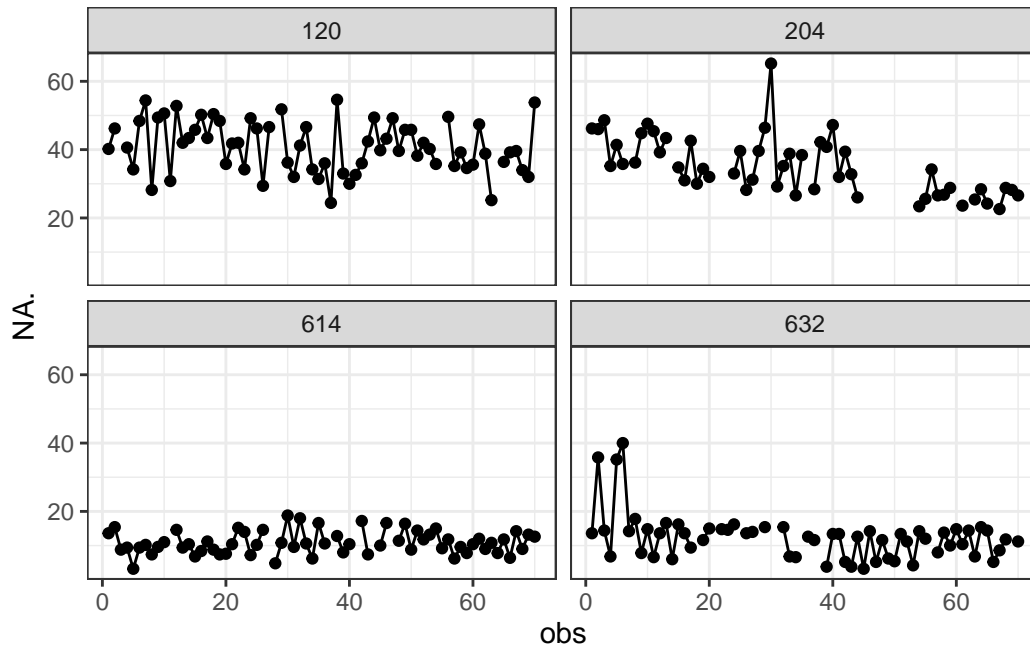
71:	637	2.2	4.60	5.8	6.520000	7.20	22.4
72:	638	0.0	6.05	9.1	9.533333	11.85	22.4
73:	640	2.8	5.40	7.2	8.114815	9.75	17.6
74:	641	0.0	2.75	8.1	9.723333	14.15	33.8
75:	642	0.0	0.00	0.5	5.451515	8.55	34.6
76:	643	4.2	8.40	11.0	12.075362	14.20	29.4
77:	644	11.0	17.20	19.7	20.696000	21.95	44.4
78:	645	5.0	9.60	12.2	17.811940	19.10	54.4
		PID	Min.	1st Qu.	Median	Mean	3rd Qu. Max.

```
# We randomly select 10 participants for plotting the
# distribution of the time-varying variable `NA.`.
n.ID.sample <- sample(unique(data$PID), 4)
data.person.sample <- data[which(data$PID %in% n.ID.sample), ]

# Histogram for the time-varying variable happy by person
ggplot(data.person.sample, aes(NA.)) +
  geom_histogram(color = "black", fill = "white", bins = 30) +
  facet_wrap(~PID) +
  scale_fill_viridis() +
  theme_bw()
```



```
# Plot the trajectories of the time-varying variable NA by person
data.person.sample %>%
  ggplot(aes(x = obs, y = NA.)) +
  geom_point() +
  geom_line() + # add lines to connect the data for each person
  facet_wrap(. ~ PID) +
  scale_fill_viridis() +
  theme_bw()
```



```
# We create a variable including the `MDD` (1 = `MDD`, 0 = control group),
# and person's means of the time-varying variable `NA.`.
dt.person <- aggregate(
  cbind(data$MDD, data$NA.),
  by = list(data$PID),
  mean,
  na.rm = TRUE
)

# Add column names.
colnames(dt.person) <- c("Group.1", "MDD", "NA.")

# See the first 6 rows.
```



```
head(dt.person)
```

	Group.1	MDD	NA.
1	101	1	29.62187
2	102	1	39.09032
3	103	1	65.52836
4	105	1	50.23529
5	108	1	17.64762
6	109	1	33.95652

```
# See the last 6 rows.  
tail(dt.person)
```

	Group.1	MDD	NA.
73	640	0	8.114815
74	641	0	9.723333
75	642	0	5.451515
76	643	0	12.075362
77	644	0	20.696000
78	645	0	17.811940

```
# Descriptive statistics for person's means of the time-varying variable `NA.`.
describe(dt.person$NA.)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	78	26.24	19.91	20.69	24.23	21.06	1.37	81.21	79.84	0.72	-0.45	2.25

```
# Descriptive statistics for person's means of the time-varying variable `NA`  
# for `MDD` = 1.  
describe(dt.person$NA.[dt.person$MDD == 1])
```

```
vars  n  mean    sd median trimmed   mad    min    max range skew kurtosis
X1     1 38 42.96 15.02  40.23   42.29 14.06 13.53 81.21 67.68 0.51    -0.06
      se
X1 2.44
```

```
# Descriptive statistics for person's means of the time-varying variable `NA.`
# for `MDD` = 0.
describe(dt.person$NA.[dt.person$MDD == 0])
```

```
vars  n  mean   sd median trimmed  mad   min    max range skew kurtosis   se
X1    1 40 10.36 6.14   9.21      9.5 4.76 1.37 29.71 28.35 1.27      1.32 0.97
```

```
# We create a variable including the `MDD` (1 = `MDD`, 0 = control group),
# and person's standard deviation of the time-varying variable `NA.`.
```

```
dt.person.sd <- aggregate(
  data$NA.,
  by = list(data$PID, data$MDD),
  sd,
  na.rm = TRUE
)
```

```
# Add column names.
```

```
colnames(dt.person.sd) <- c("Group.1", "MDD", "NA.")
```

```
# See the first 6 rows.
```

```
head(dt.person.sd)
```

	Group.1	MDD	NA.
1	601	0	10.139970
2	602	0	2.905915
3	603	0	12.948615
4	604	0	6.165745
5	606	0	3.925499
6	607	0	12.595571

```
# See the last 6 rows.
```

```
tail(dt.person.sd)
```

	Group.1	MDD	NA.
73	206	1	10.115789
74	207	1	12.116928
75	208	1	12.489439
76	210	1	17.420019
77	211	1	6.747881
78	306	1	14.098726

```
# Descriptive statistics for person's standard deviation of the time-varying
# variable `NA.`.
describe(dt.person.sd$NA.)
```

```
vars  n mean  sd median trimmed  mad  min   max range skew kurtosis  se
X1    1 78 8.97 4.67   8.67    8.56 5.22 2.14 27.32 25.18 1.06    1.81 0.53
```

```
# Descriptive statistics for person's standard deviation of the time-varying
# variable `NA.` for `MDD` = 1.
describe(dt.person.sd$NA.[dt.person.sd$MDD == 1])
```

```
vars  n mean  sd median trimmed  mad  min   max range skew kurtosis  se
X1    1 38 11.44 4.7  10.98    11 3.59 4.16 27.32 23.16 1.1    1.76 0.76
```

```
# Descriptive statistics for person's standard deviation of the time-varying
# variable `NA.` for `MDD` = 0.
describe(dt.person.sd$NA.[dt.person.sd$MDD == 0])
```

```
vars  n mean  sd median trimmed  mad  min   max range skew kurtosis  se
X1    1 40 6.63 3.24   5.62    6.35 3.29 2.14 12.95 10.8 0.62   -0.89 0.51
```

Example 1

Estimating the effect of a continuous time-varying predictor

The first illustrative example shows how to estimate the effect of a time-varying predictor on the outcome of interest. Considering the Leuven clinical study, we are interested in studying the impact of *anhedonia* on *negative affect* in daily life on patients with *major depressive disorder*.

We use the data of 38 individuals diagnosed with MDD. We select the individuals diagnosed with MDD.

```
# Create `MDD` subset.
data.MDD <- data[which(data$MDD == 1), ]
```

First, we are going to estimate the individual means, the mean across all participants, and the standard deviation of the variable *anhedonia*.

```

# Compute the group mean of anhedonia.
groupmean_X = aggregate(
  data.MDD$anhedonia,
  list(data.MDD$PID),
  FUN = mean,
  data = data.MDD,
  na.rm = TRUE
)

# Compute the mean.
mean_X <- mean(groupmean_X[, 2])

# Print the mean.
print(mean_X)

```

```
[1] 51.66162
```

```

# Compute the standard deviation.
sd_X <- sd(data.MDD$anhedonia, na.rm = TRUE)

# Print the standard deviation.
print(sd_X)

```

```
[1] 23.6734
```

Next, we are going to person mean-centered the variable `anhedonia`.

```

# Centered within individuals anhedonia.
N.i <- unique(data.MDD$PID)
anhedonia.c = rep(0, nrow(data.MDD))

for (i in N.i) {
  # Get the anhedonia for the i-th individual.
  ith_anhedonia <- data.MDD$anhedonia[which(data.MDD$PID == i)]

  # Get the mean of anhedonia for the i-th individual.
  ith_anhedonia_mean <- mean(data.MDD$anhedonia[which(data.MDD$PID == i)], na.rm = TRUE)

  # Center.
  anhedonia.c[which(data.MDD$PID == i)] <- ith_anhedonia - ith_anhedonia_mean
}

```

```

}

# Add the centered variable to the data.
data.MDD <- cbind(data.MDD, anhedonia.c)

```

We estimate the linear mixed-effects model assuming AR(1) errors:

```

# Fit a linear mixed-effects model to data.
fit.Model.1 = lme(
  fixed = NA. ~ 1 + anhedonia.c,
  random = ~ 1 + anhedonia.c | PID,
  na.action = na.omit,
  data = data.MDD,
  correlation = corAR1(),
  method = "REML"
)

```

The summary of the estimation results is given by:

```

# Summary of the estimation results.
summary(fit.Model.1)

```

Linear mixed-effects model fit by REML

```

Data: data.MDD
      AIC      BIC    logLik
17319.52 17359.56 -8652.758

```

Random effects:

```

Formula: ~1 + anhedonia.c | PID
Structure: General positive-definite, Log-Cholesky parametrization
              StdDev      Corr
(Intercept) 14.7788373 (Intr)
anhedonia.c  0.1162717 0.003
Residual    11.9150994

```

Correlation Structure: AR(1)

```

Formula: ~1 | PID
Parameter estimate(s):
      Phi
0.4293834

```

```

Fixed effects: NA. ~ 1 + anhedonia.c
              Value Std.Error   DF   t-value p-value

```

```
(Intercept) 42.98279 2.4299657 2216 17.688641      0
anhedonia.c 0.13900 0.0233386 2216  5.955752      0
Correlation:
      (Intr)
anhedonia.c 0.002
```

```
Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-4.21865272 -0.56951096 -0.04394045  0.53168368  6.16917066
```

```
Number of Observations: 2255
Number of Groups: 38
```

Obtain confidence intervals:

```
# Confidence intervals.
intervals(fit.Model.1, which = "fixed")
```

Approximate 95% confidence intervals

```
Fixed effects:
      lower      est.      upper
(Intercept) 38.21754213 42.9827900 47.7480379
anhedonia.c  0.09323107  0.1389989  0.1847667
```

The estimated fixed intercept is given by:

```
# Extract fixed effect coefficients.
# Extract the value of fixed intercept.
coef(summary(fit.Model.1))[1, 1]
```

```
[1] 42.98279
```

the effect of the level 2 continuous variable on the intercept is extracted as follows:

```
# Extract the value of the fixed slope.
coef(summary(fit.Model.1))[2, 1]
```

```
[1] 0.1389989
```

The standard deviation and autocorrelation of the level 1 residuals are extracted as follows:

```
# Extract level 1 residuals standard deviation.  
as.numeric(VarCorr(fit.Model.1)[3, 2])
```

```
[1] 11.9151
```

```
# Extract level 1 residuals correlation between consecutive points  
as.numeric(coef(  
  fit.Model.1$modelStruct$corStruct,  
  unconstrained = FALSE  
))
```

```
[1] 0.4293834
```

The standard deviation of the random intercept is given by:

```
# Extract random effect covariance structure.  
# Extract the standard deviation of the random intercept.  
as.numeric(VarCorr(fit.Model.1)[1, 2])
```

```
[1] 14.77884
```

The standard deviation of the random slope is given by:

```
# Extract random effect covariance structure.  
# Extract the standard deviation of the random slope.  
as.numeric(VarCorr(fit.Model.1)[2, 2])
```

```
[1] 0.1162717
```

The correlation between the random intercept and the random slope is given by:

```
# Extract random effect covariance structure.  
# Extract the standard deviation of the random slope.  
as.numeric(VarCorr(fit.Model.1)[2, 3])
```

```
[1] 0.003
```

Example 2

Estimating cross-level interaction effect between a continuous time-varying predictor and a continuous time-invariant predictor

We now show how to estimate a cross-level interaction effect between a continuous time-varying predictor and continuous time-invariant predictor. In particular, we are interested in studying if depression moderated the impact of anhedonia on negative affect in daily life for individuals diagnosed with MDD.

Before estimating the model, we are going to compute the mean and standard deviation of the level 2 variable QIDS.

```
# Compute the mean of `W`.
groupmean_W = aggregate(
  data.MDD$QIDS,
  list(data.MDD$PID),
  FUN = mean,
  data = data.Controls,
  na.rm = TRUE,
  method = "REML"
)

# Compute the mean.
mean_W <- mean(groupmean_W[, 2])

# Print the mean.
print(mean_W)
```

```
[1] 15.71053
```

```
# Compute the standard deviation.
sd_W <- sd(groupmean_W[, 2])

# Print the standard deviation.
print(sd_W)
```

```
[1] 4.996798
```

Next, we are going to mean centered the variable QIDS using the mean estimated above:


```

# Centered QIDS.
N.i <- unique(data.MDD$PID)
QIDS.c <- rep(0, nrow(data.MDD))

# For each participant.
for (i in N.i) {
  # Extract the value of the variable for the i-th individual.
  ith_QIDS <- data.MDD$QIDS[which(data.MDD$PID == i)]

  # Center the variable.
  QIDS.c[which(data.MDD$PID == i)] <- ith_QIDS - mean_W
}

# Add the centered variable to the data.
data.MDD <- cbind(data.MDD, QIDS.c)

```

Next, we estimate the linear mixed-effects model assuming AR(1) errors:

```

# Fit a linear mixed-effects model to data.
fit.Model.2 = lme(
  fixed = NA. ~ 1 + anhedonia.c + anhedonia.c * QIDS.c,
  random = ~ 1 + anhedonia.c | PID,
  na.action = na.omit,
  data = data.MDD,
  correlation = corAR1(),
  method = "REML"
)

```

The summary of the estimation results is given by:

```

# Print the summary of the estimation results.
summary(fit.Model.2)

```

Linear mixed-effects model fit by REML

```

Data: data.MDD
      AIC      BIC    logLik
17315.42 17366.89 -8648.708

```

Random effects:

```

Formula: ~1 + anhedonia.c | PID
Structure: General positive-definite, Log-Cholesky parametrization
      StdDev      Corr

```

```

(Intercept) 12.855527 (Intr)
anhedonia.c 0.105615 0.249
Residual    11.923407

```

Correlation Structure: AR(1)

Formula: ~1 | PID

Parameter estimate(s):

Phi

0.430249

Fixed effects: NA. ~ 1 + anhedonia.c + anhedonia.c * QIDS.c

	Value	Std.Error	DF	t-value	p-value
(Intercept)	42.97796	2.1228674	2215	20.245238	0.0000
anhedonia.c	0.13747	0.0218390	2215	6.294570	0.0000
QIDS.c	1.52600	0.4308467	36	3.541864	0.0011
anhedonia.c:QIDS.c	-0.01019	0.0046382	2215	-2.197917	0.0281

Correlation:

	(Intr)	anhdn.	QIDS.c
anhedonia.c	0.191		
QIDS.c	-0.001	0.000	
anhedonia.c:QIDS.c	0.000	-0.031	0.183

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-4.20450995	-0.56155161	-0.03764376	0.53518590	6.15760197

Number of Observations: 2255

Number of Groups: 38

Obtain confidence intervals:

```

# Print the confidence intervals.
intervals(fit.Model.2, which = "fixed")

```

Approximate 95% confidence intervals

Fixed effects:	lower	est.	upper
(Intercept)	38.81493867	42.97795723	47.140975796
anhedonia.c	0.09464004	0.13746708	0.180294127
QIDS.c	0.65220263	1.52600019	2.399797755
anhedonia.c:QIDS.c	-0.01929006	-0.01019438	-0.001098702

The estimated fixed intercept is given by:

```
# Extract fixed effect coefficients.  
# Extract the value of fixed intercept.  
coef(summary(fit.Model.2))[1, 1]
```

```
[1] 42.97796
```

The effect of the level 2 continuous variable on the intercept is extracted as follows:

```
# Extract the value of the fixed slope.  
coef(summary(fit.Model.2))[2, 1]
```

```
[1] 0.1374671
```

The effect of the level 2 continuous variable on the intercept is extracted as follows:

```
# Extract the value of the fixed slope.  
coef(summary(fit.Model.2))[3, 1]
```

```
[1] 1.526
```

The effect of the level 2 continuous variable on the intercept is extracted as follows:

```
# Extract the value of the fixed slope.  
coef(summary(fit.Model.2))[4, 1]
```

```
[1] -0.01019438
```

The standard deviation and autocorrelation of the level 1 residuals are extracted as follows:

```
# Extract level 1 residuals standard deviation.  
as.numeric(VarCorr(fit.Model.2)[3, 2])
```

```
[1] 11.92341
```

```
# Extract level 1 residuals correlation between consecutive points.
as.numeric(coef(
  fit.Model.2$modelStruct$corStruct,
  unconstrained = FALSE
))
```

```
[1] 0.430249
```

The standard deviation of the random intercept is given by:

```
# Extract random effect covariance structure.
# Extract the standard deviation of the random intercept.
as.numeric(VarCorr(fit.Model.2)[1, 2])
```

```
[1] 12.85553
```

The standard deviation of the random slope is given by:

```
# Extract random effect covariance structure.
# Extract the standard deviation of the random slope.
as.numeric(VarCorr(fit.Model.2)[2, 2])
```

```
[1] 0.105615
```

The correlation between the random intercept and the random slope is given by:

```
# Extract random effect covariance structure.
# Extract the standard deviation of the random slope.
as.numeric(VarCorr(fit.Model.2)[2, 3])
```

```
[1] 0.249
```

Example 3

Estimate group differences in the autoregressive effect in multilevel AR(1) models

In this illustration, we are interested in estimating differences in the autoregressive effect of negative affect between participants diagnosed with major depressive disorder (MDD) and

control subjects. The dataset contains 38 participants diagnosed with MDD and 40 control subjects.

First, for each individual, we are going to compute the lagged variable negative affect (i.e., NA.). The variable negative affect is lagged within each day.

```
# Create a lag variable.
# The data is lag within a person and days.
NA.lag <- rep(0, nrow(data))
subjno.i <- unique(data$PID)

# For each subject.
for (i in subjno.i) {
  n.i = which(data$PID == i)
  Day.i = data$day[n.i]

  # For each day.
  for (t in unique(Day.i)) {
    k.i = n.i[which(data$day[n.i] == t)]
    NA.lag[k.i] = shift(data$NA.[k.i], 1)
  }
}

# Add the lagged variable to the data.
data <- cbind(data, NA.lag)
```

The lagged variable NA.lag will be centered using the individual's mean.

```
# Centered within individuals NA.lag.
N.i <- unique(data$PID)
NA.lag.c <- rep(0, nrow(data))

# For each individual.
for (i in N.i) {
  # Get the `NA.lag` for the i-th individual.
  ith_na_lag <- data$NA.lag[which(data$PID == i)]

  # Get the `NA.lag` mean for the i-th individual.
  ith_na_lag_mean <- mean(data$NA.[which(data$PID == i)], na.rm = TRUE)

  # Center.
  NA.lag.c[which(data$PID == i)] <- ith_na_lag - ith_na_lag_mean
}
```

```

}

# Add the centered lagged variable to the data.
data <- cbind(data, NA.lag.c)

```

To estimate the model, we use the function `lme` from the `nlme` R package. The dependent variable is the negative affect (i.e. NA.), the predictor is the lagged outcome, which is centered using the individuals' mean:

```

# Fit a linear mixed-effects model to data.
fit.Model.3 <- lme(
  fixed = NA. ~ 1 + MDD + NA.lag + MDD * NA.lag,
  random = ~ 1 + NA.lag | PID,
  na.action = na.omit,
  data = data,
  method = "REML"
)

```

where NA. is the negative affect, 1 is the fixed intercept, MDD is the difference in the fixed intercept between the two groups, NA.lag.c is the fixed autoregressive effect and MDD*NA.lag.c is the difference in the fixed autoregressive effect between the two groups. The random effect structure of the model is 1 + NA.lag.c|PID, where 1 is the random intercept, and NA.lag.c is the random slope, which is allowed to vary over participants (PID).

The summary of the estimation results is given by:

```

# Print the summary of the model.
summary(fit.Model.3)

```

Linear mixed-effects model fit by REML

```

Data: data
      AIC      BIC    logLik
28968.54 29018.86 -14476.27

```

Random effects:

```

Formula: ~1 + NA.lag | PID
Structure: General positive-definite, Log-Cholesky parametrization
      StdDev   Corr
(Intercept) 5.7874515 (Intr)
NA.lag      0.1402728 -0.199
Residual    8.7540299

```

```
Fixed effects: NA. ~ 1 + MDD + NA.lag + MDD * NA.lag
              Value Std.Error   DF   t-value p-value
(Intercept)  6.824841 0.9800415 3911   6.963828  0.0000
MDD          16.326601 1.5896208   76  10.270752  0.0000
NA.lag       0.313887 0.0366665 3911   8.560571  0.0000
MDD:NA.lag   0.116184 0.0472240 3911   2.460275  0.0139
Correlation:
      (Intr) MDD   NA.lag
MDD      -0.617
NA.lag   -0.339  0.209
MDD:NA.lag 0.263 -0.417 -0.776
```

```
Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-5.5027244 -0.4822533 -0.1062090  0.3948620  6.3499125
```

```
Number of Observations: 3991
Number of Groups: 78
```

We extract the estimated fixed intercept as follows,

```
# Extract fixed effect coefficients.
# Extract the value of fixed intercept.
coef(summary(fit.Model.3))[1, 1]
```

```
[1] 6.824841
```

The differences on the intercept between the two groups is given by:

```
# Extract the value of the difference in the fixed intercept between the two
# groups.
coef(summary(fit.Model.3))[2, 1]
```

```
[1] 16.3266
```

The fixed autoregressive effect is:

```
# Extract the value of fixed slope.
coef(summary(fit.Model.3))[3, 1]
```

```
[1] 0.3138866
```

And the difference in the autoregressive effect between the two groups is extracted as follows:

```
# Extract the value of the difference in the fixed slope between  
# the two groups.  
coef(summary(fit.Model.3))[4, 1]
```

```
[1] 0.1161839
```

The standard deviation of the level 1 residuals is extracted as follows:

```
# Extract level 1 residuals standard deviation.  
as.numeric(VarCorr(fit.Model.3)[3, 2])
```

```
[1] 8.75403
```

The standard deviation of the random intercept is given by:

```
# Extract random effect covariance structure.  
# Extract the standard deviation of the random intercept.  
as.numeric(VarCorr(fit.Model.3)[1, 2])
```

```
[1] 5.787452
```

The standard deviation of the random slope is given by:

```
# Extract random effect covariance structure.  
# Extract the standard deviation of the random slope.  
as.numeric(VarCorr(fit.Model.3)[2, 2])
```

```
[1] 0.1402728
```

The correlation between the random intercept and the random slope is given by:

```
# Extract random effect covariance structure.  
# Extract the standard deviation of the random slope.  
as.numeric(VarCorr(fit.Model.3)[2, 3])
```

```
[1] -0.199
```


Session information

Using the command below, we can print the **session** information (i.e., operating system, details about the R installation, and so on) for reproducibility purposes.

```
# Session information.  
sessionInfo()
```

```
R version 4.3.0 (2023-04-21)  
Platform: aarch64-apple-darwin20 (64-bit)  
Running under: macOS Ventura 13.4
```

```
Matrix products: default
```

```
BLAS: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
```

```
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;
```

```
locale:
```

```
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

```
time zone: Europe/Amsterdam
```

```
tzcode source: internal
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] nlme_3.1-162      viridis_0.6.3      viridisLite_0.4.2 psych_2.3.3  
[5] data.table_1.14.8 lubridate_1.9.2     forcats_1.0.0      stringr_1.5.0  
[9] dplyr_1.1.2       purrr_1.0.1        readr_2.1.4        tidyr_1.3.0  
[13] tibble_3.2.1      ggplot2_3.4.2      tidyverse_2.0.0
```

```
loaded via a namespace (and not attached):
```

```
[1] utf8_1.2.3        generics_0.1.3     stringi_1.7.12     lattice_0.21-8  
[5] hms_1.1.3         digest_0.6.31      magrittr_2.0.3     evaluate_0.21  
[9] grid_4.3.0        timechange_0.2.0   fastmap_1.1.1      jsonlite_1.8.5  
[13] gridExtra_2.3     fansi_1.0.4        scales_1.2.1       mnormt_2.1.1  
[17] cli_3.6.1         rlang_1.1.1        munsell_0.5.0      withr_2.5.0  
[21] yaml_2.3.7        tools_4.3.0        parallel_4.3.0     tzdb_0.4.0  
[25] colorspace_2.1-0  vctrs_0.6.2        R6_2.5.1           lifecycle_1.0.3  
[29] pkgconfig_2.0.3   pillar_1.9.0       gtable_0.3.3       glue_1.6.2  
[33] xfun_0.39         tidyselect_1.2.0   rstudioapi_0.14    knitr_1.43  
[37] farver_2.1.1      htmltools_0.5.5    labeling_0.4.2     rmarkdown_2.22
```

[41] compiler_4.3.0

References

- Heininga, V. E., Dejonckheere, E., Houben, M., Obbels, J., Sienaert, P., Leroy, B., Roy, J. van, & Kuppens, P. (2019). The dynamical signature of anhedonia in major depressive disorder: Positive emotion dynamics, reactivity, and recovery. *BMC Psychiatry*, 19(1), 59.
- Rush, A. J., Trivedi, M. H., Ibrahim, H. M., Carmody, T. J., Arnow, B., Klein, D. N., Markowitz, J. C., Ninan, P. T., Kornstein, S., Manber, R., et al. (2003). The 16-item quick inventory of depressive symptomatology (QIDS), clinician rating (QIDS-c), and self-report (QIDS-SR): A psychometric evaluation in patients with chronic major depression. *Biological Psychiatry*, 54(5), 573–583.