

# Sample size solutions for $N = 1$ intensive longitudinal designs

Ginette Lafit

Jordan Revol

2023-06-02

## Table of contents

Setup the environment . . . . .	1
The Leuven clinical study data set . . . . .	2
Visualizations and descriptive statistics . . . . .	5
1. Estimate the AR(1) model for PA . . . . .	11
2. Estimate the VAR(1) model for PA and NA . . . . .	12
3. Estimate the VAR(1) model for NA, PA and Anhedonia . . . . .	13
How to run the shiny app? . . . . .	16
Exercise: Power analysis of VAR(1) with 3 variables . . . . .	16
Power analysis result . . . . .	16
Sensitivity analysis for power: varying parameters . . . . .	17
Sensitivity analysis for power: using CI . . . . .	17
Exercise: PAA of VAR(1) with 3 variables . . . . .	18
PAA result . . . . .	18
Sensitivity analysis for power: varying parameters . . . . .	20
Get the session info . . . . .	20

## Setup the environment

The code below chunk simply makes sure that all the libraries used here are installed. We should first check if the R packages are installed before we proceed.

```
## Do not run because we do not want to install packages (this should be your decision)

list.of.packages = c("data.table","psych","ggplot2","tidyverse","MASS")
new.packages = list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
```

```
if(length(new.packages)) install.packages(new.packages)
```

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

```
library(data.table) # to create lagged outcome
library(psych) # to compute descriptive statistics
library(ggplot2) # for making plots
library(tidyverse) # a useful package

library(MASS)

set.seed(1235) # Set a seed to reproduce analyses
```

## The Leuven clinical study data set

We use data from Heininga et al. (2019); this study applies the ESM methodology to study emotion dynamics in people with Major Depressive Disorder. The study consist of an ESM testing period of 7 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.

First, we are going to load the data set:

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

Now, we are going to explore the data set to get a better understanding of the what's inside.

```
# Select the first participant diagnosed with major depressive
i.ID = unique(data$PID[data$MDD==1])[1]

# Select data from participant with person identification number PID=101
data = data[data$PID==101,]

# Find the dimensions
dim(data)
```

```
[1] 70 8
```

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

```
'data.frame': 70 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
65	101	7	5	24.33333	31.8	52	1	12
66	101	7	6	28.66667	20.6	53	1	12
67	101	7	7	23.33333	23.8	51	1	12
68	101	7	8	33.66667	36.2	46	1	12
69	101	7	9	41.66667	21.0	29	1	12
70	101	7	10	34.00000	18.4	47	1	12

```
# 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	Min.	:1	Min.	: 1.0	Min.	:14.67	Min.	:14.40
1st Qu.:	:101	1st Qu.:	:2	1st Qu.:	: 3.0	1st Qu.:	:26.08	1st Qu.:	:21.40
Median	:101	Median	:4	Median	: 5.5	Median	:33.33	Median	:25.90
Mean	:101	Mean	:4	Mean	: 5.5	Mean	:33.51	Mean	:29.62
3rd Qu.:	:101	3rd Qu.:	:6	3rd Qu.:	: 8.0	3rd Qu.:	:41.67	3rd Qu.:	:36.25
Max.	:101	Max.	:7	Max.	:10.0	Max.	:57.33	Max.	:65.60
						NA's	:6	NA's	:6

anhedonia		MDD		QIDS	
Min.	:14.00	Min.	:1	Min.	:12
1st Qu.:	:24.75	1st Qu.:	:1	1st Qu.:	:12
Median	:41.50	Median	:1	Median	:12
Mean	:39.34	Mean	:1	Mean	:12
3rd Qu.:	:52.00	3rd Qu.:	:1	3rd Qu.:	:12
Max.	:83.00	Max.	:1	Max.	:12
NA's	:6				

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

```
[1] 1
```

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?’ and ‘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?’ and ‘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, finally QIDS denotes the sum of the items of the Quick Inventory of Depressive Symptomatology (i.e. QIDS) [rush200316].

QIDS was measured before the ESM testing period. Time-varying variables (PA, NA, and anhedonia) have been lagged within days to account for the night breaks.

## Visualizations and descriptive statistics

We first obtain some descriptive statistics including number of observations per day, and compliance.

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

```
101
70
```

```
# 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 for the participant PID=101
describe(data$Compliance)
```

```
vars  n mean  sd median trimmed mad min max range skew kurtosis  se
X1    1 70 0.91 0.28      1      1  0  0  1      1 -2.9      6.48 0.03
```

Next, we can obtain visualizations and statistics of the distribution of the person-level or time-invariant variables

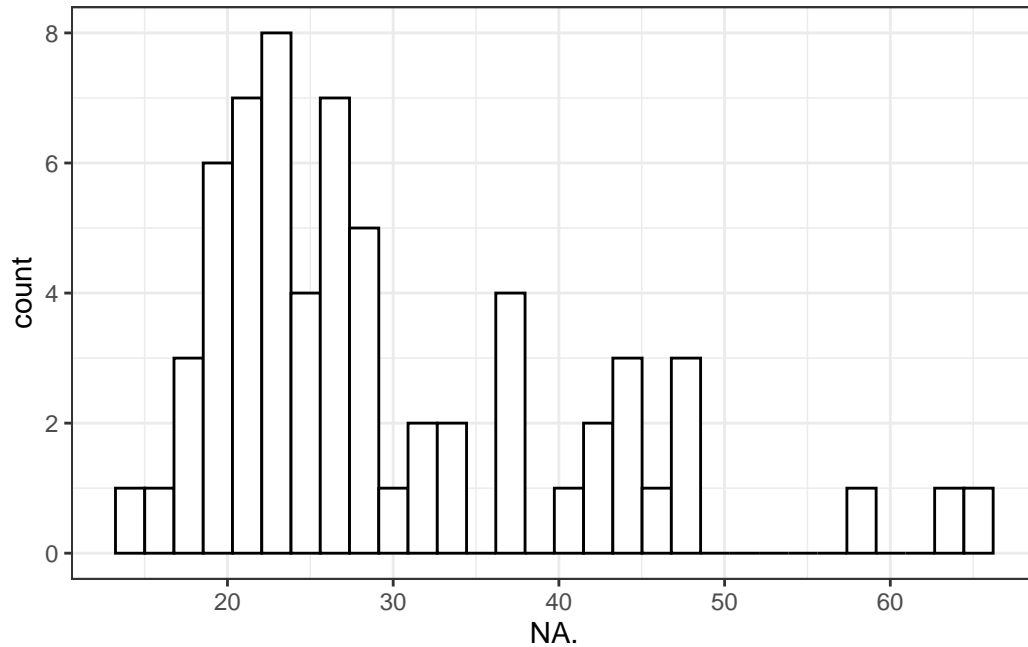
```
# We create a data set that will aggregate the data from the time invariant
# variables: diagnosis (1 = MDD, 0 = control) and depression (QIDS)
dt.person = aggregate(cbind(data$MDD,data$QIDS), by = list(data$PID), mean, na.rm = TRUE)
colnames(dt.person) = c("Group.1","MDD","QIDS")
dt.person
```

```
Group.1 MDD QIDS
1      101  1   12
```

We now focus on time-varying variables NA, PA, and anhedonia and we obtain visualization and descriptive statistics

```
# Histogram for the time-varying variable negative affect (NA.)
ggplot(data, aes(NA.)) + geom_histogram(color="black", fill="white",bins=30) + theme_bw()
```

Warning: Removed 6 rows containing non-finite values (`stat\_bin()`).



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

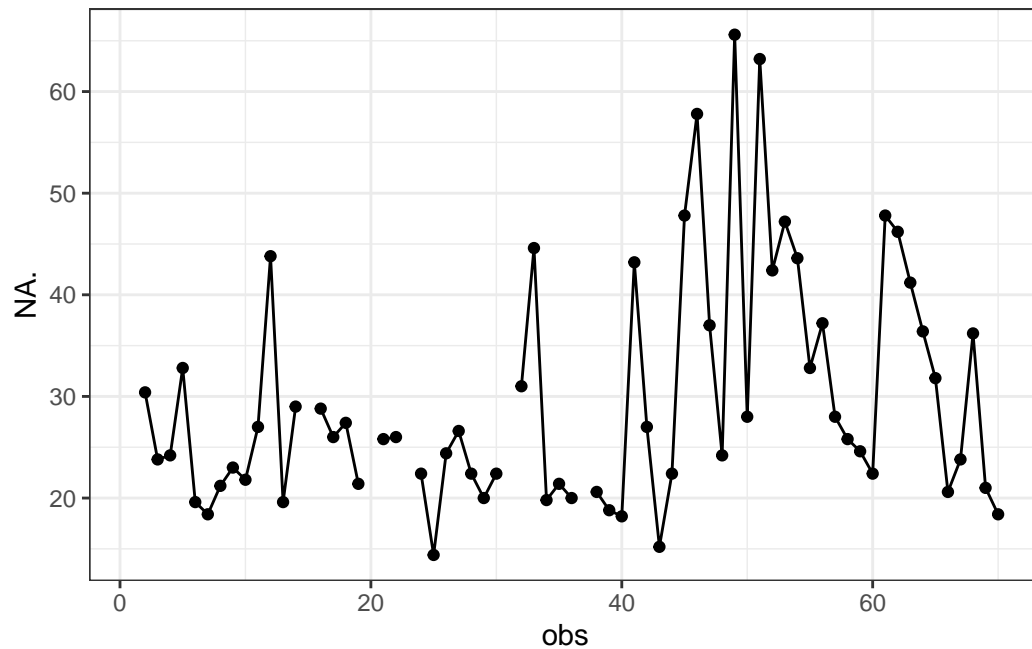
```
vars n mean sd median trimmed mad min max range skew kurtosis se
X1 1 64 29.62 11.47 25.9 28.13 7.86 14.4 65.6 51.2 1.25 1.01 1.43
```

```
# Create obs order variable
data$obs = 1:nrow(data)

# Plot the trajectories of the time-varying variable NA by person
data %>%
  ggplot(aes(x = obs, y = NA.)) +
  geom_point() +
  geom_line() + theme_bw()
```

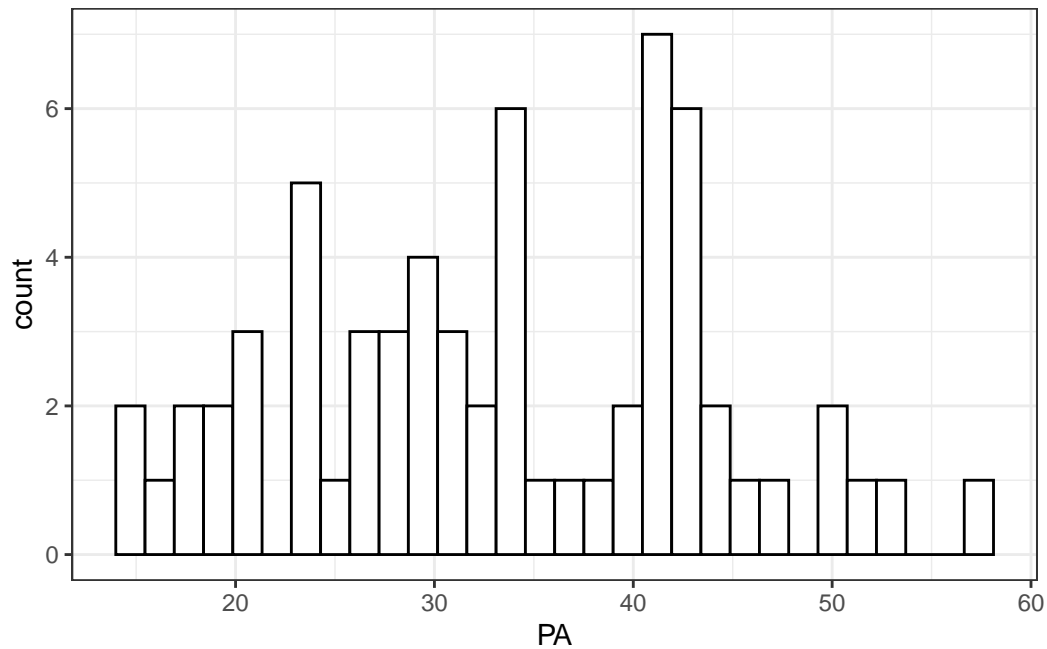
Warning: Removed 6 rows containing missing values (``geom_point()``).

Warning: Removed 1 row containing missing values (``geom_line()``).



```
# Histogram for the time-varying variable negative affect (PA)
ggplot(data, aes(PA)) + geom_histogram(color="black", fill="white",bins=30) + theme_bw()
```

Warning: Removed 6 rows containing non-finite values (``stat_bin()``).



```
# Descriptive statistics for PA
describe(data$PA)
```

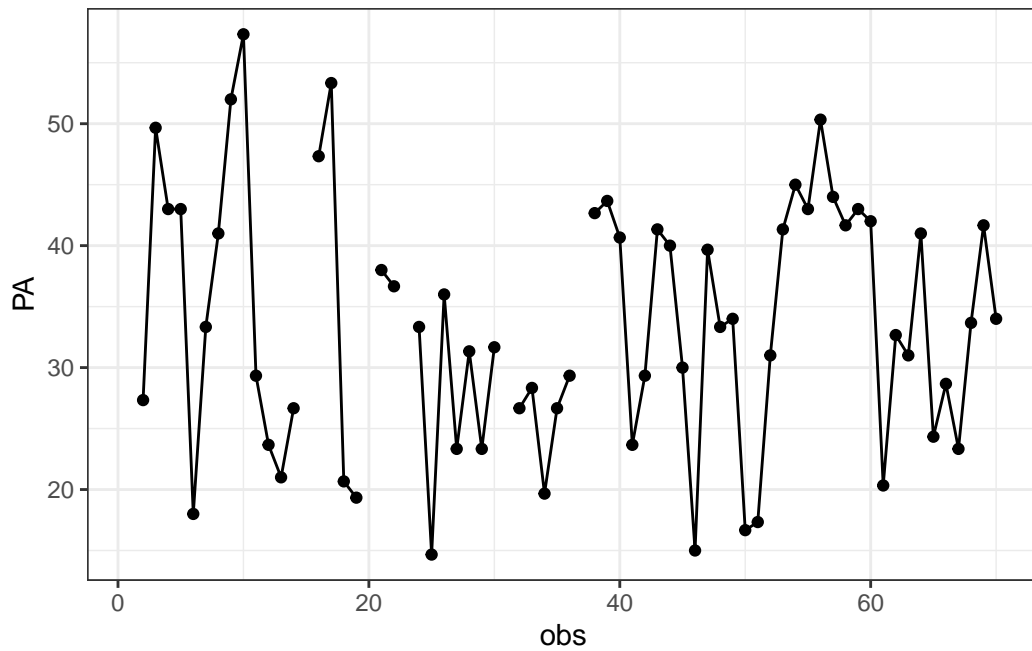
```
vars  n  mean   sd median trimmed  mad   min   max range skew kurtosis
X1    1  64 33.51 10.32  33.33   33.33 12.35 14.67 57.33 42.67 0.09   -0.84
se
X1  1.29
```

```
# Plot the trajectories of the time-varying variable PA by person
data %>%
  ggplot(aes(x = obs, y = PA)) +
  geom_point() +
  geom_line() + theme_bw()
```

Warning: Removed 6 rows containing missing values (`geom\_point()`).

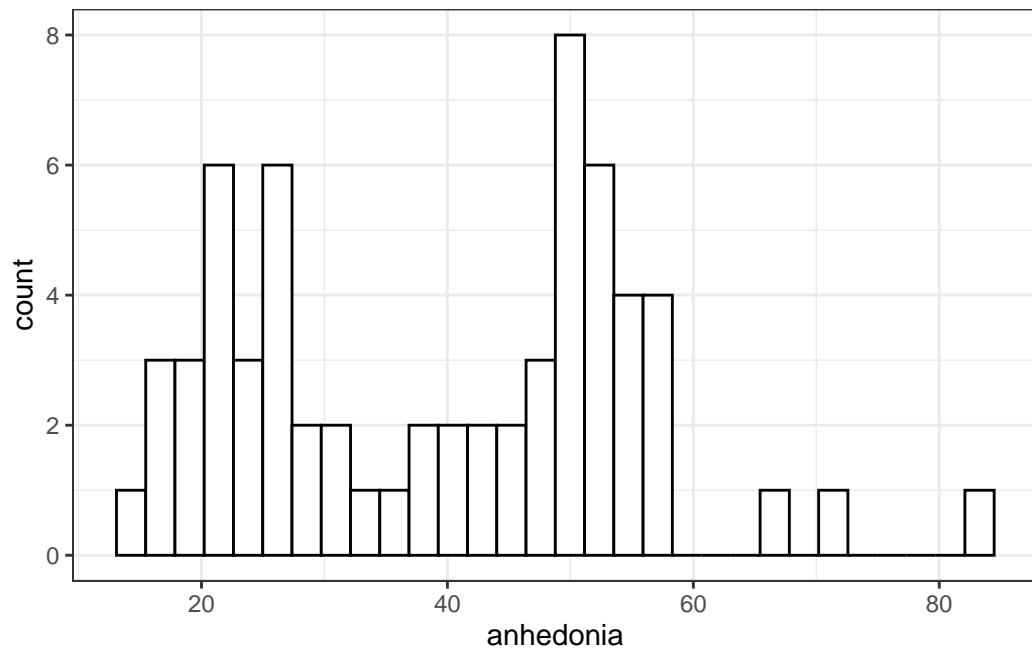
Removed 1 row containing missing values (`geom\_line()`).





```
# Histogram for the time-varying variable anhedonia
ggplot(data, aes(anhedonia)) + geom_histogram(color="black", fill="white",bins=30) + theme
```

Warning: Removed 6 rows containing non-finite values (`stat\_bin()`).

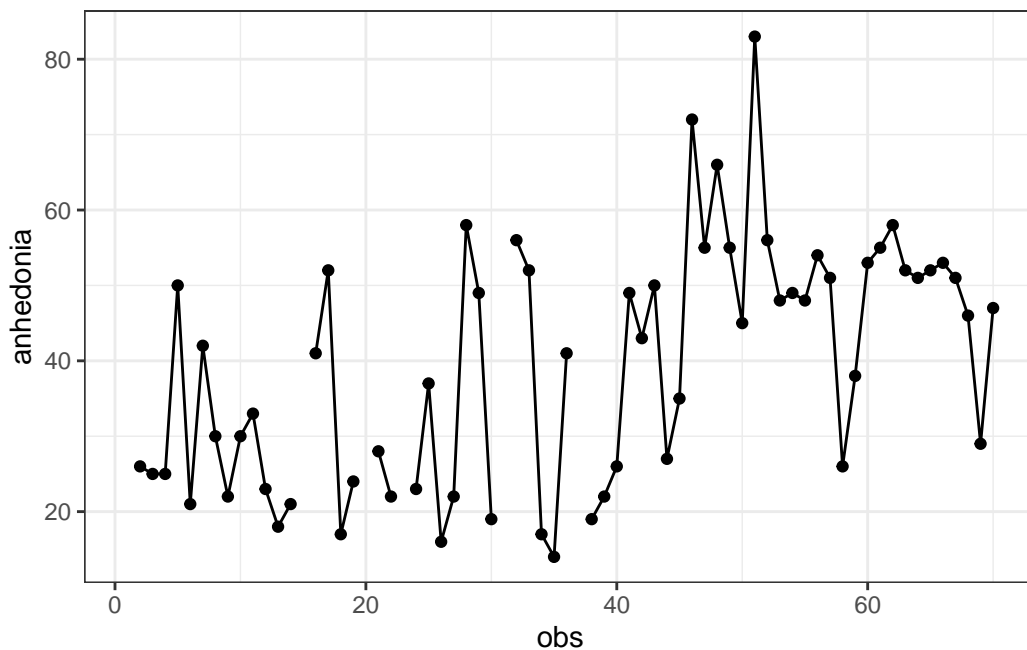


```
# Descriptive statistics for anhedonia
describe(data$anhedonia)
```

```
vars  n  mean    sd median trimmed  mad min max range skew kurtosis  se
X1    1 64 39.34 15.84  41.5   38.92 20.02  14 83   69 0.22   -0.77 1.98
```

```
# Plot the trajectories of the time-varying variable anhedonia by person
data %>%
  ggplot(aes(x = obs, y = anhedonia)) +
  geom_point() +
  geom_line() + theme_bw()
```

Warning: Removed 6 rows containing missing values (`geom\_point()`).  
 Removed 1 row containing missing values (`geom\_line()`).



Finally, we create the lagged variables for PA and NA. They will be used on the following AR(1) and VAR(1) models.

```
# Create lagged variables: lagged within days to take into account night breaks
data$PA.lag = rep(NA, nrow(data))
```

```

data$NA.lag = rep(NA,nrow(data))
data$anhedonia.lag = rep(NA,nrow(data))
day.id = unique(data$day)
for (t in day.id){
  data$PA.lag[which(data$day==t)] = shift(data$PA[which(data$day==t)],1)
  data$NA.lag[which(data$day==t)] = shift(data$NA.[which(data$day==t)],1)
  data$anhedonia.lag[which(data$day==t)] = shift(data$anhedonia[which(data$day==t)],1)
}

```

## 1. Estimate the AR(1) model for PA

We estimate an AR(1) model for PA using a linear regression model (ordinary least squares, OLS). You can extract the estimates with the 'summary()' function. Finally, you can compute the estimate of the standard deviation of the errors of the AR(1) model computing the standard deviation using the function 'sd()' on the residuals of the fitted model.

```

# AR(1) model for PA
fit.AR.PA = lm(PA ~ 1 + PA.lag, data = data)
summary(fit.AR.PA)

```

Call:

```
lm(formula = PA ~ 1 + PA.lag, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-21.486	-5.866	2.070	5.820	18.155

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	20.3258	4.5755	4.442	4.68e-05 ***
PA.lag	0.4092	0.1308	3.130	0.00287 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.658 on 52 degrees of freedom

(16 observations deleted due to missingness)

Multiple R-squared: 0.1585, Adjusted R-squared: 0.1423

F-statistic: 9.796 on 1 and 52 DF, p-value: 0.002866

```
# Estimate the standard deviation of the errors
sd(residuals(fit.AR.PA))
```

```
[1] 9.566865
```

## 2. Estimate the VAR(1) model for PA and NA

We estimate a VAR(1) model for PA and NA using two separate linear regression models. You can extract the estimates with the ‘summary()’ function. Finally, you can compute the estimate of the variance-covariance matrix of the errors of the VAR(1) model computing the covariance matrix using the function ‘cov()’ on the residuals of each of the fitted models.

```
# Linear regression model for PA
fit.VAR.PA = lm(PA ~ 1 + PA.lag + NA.lag, data = data)
summary(fit.VAR.PA)
```

Call:

```
lm(formula = PA ~ 1 + PA.lag + NA.lag, data = data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-21.551	-6.480	1.262	5.859	18.008

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	23.45530	6.27640	3.737	0.000471 ***
PA.lag	0.39213	0.13341	2.939	0.004930 **
NA.lag	-0.08273	0.11299	-0.732	0.467416

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.702 on 51 degrees of freedom

(16 observations deleted due to missingness)

Multiple R-squared: 0.1673, Adjusted R-squared: 0.1346

F-statistic: 5.123 on 2 and 51 DF, p-value: 0.009391

```
# Linear regression model for NA
fit.VAR.NA = lm(NA. ~ 1 + PA.lag + NA.lag, data = data)
summary(fit.VAR.NA)
```

Call:

```
lm(formula = NA. ~ 1 + PA.lag + NA.lag, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-14.260	-5.933	-2.073	3.104	39.302

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	17.8818	6.6337	2.696	0.00949 **
PA.lag	-0.0202	0.1410	-0.143	0.88664
NA.lag	0.3756	0.1194	3.145	0.00277 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.25 on 51 degrees of freedom

(16 observations deleted due to missingness)

Multiple R-squared: 0.1692, Adjusted R-squared: 0.1366

F-statistic: 5.194 on 2 and 51 DF, p-value: 0.008851

```
# Estimate variance-covariance matrix of the errors
res = cbind(residuals(fit.VAR.PA),residuals(fit.VAR.NA))
cov(res)
```

	[,1]	[,2]
[1,]	90.572865	4.295723
[2,]	4.295723	101.177796

### 3. Estimate the VAR(1) model for NA, PA and Anhedonia

We estimate a VAR(1) model for PA, NA and anhedonia using three separate linear regression models. You can extract the estimates with the 'summary()' function. Finally, you can compute the estimate of the variance-covariance matrix of the errors of the VAR(1) model by computing the covariance matrix using the function 'cov()' on the residuals of each of the fitted models.

```
# Linear regression model for PA
fit.VAR.PA = lm(PA ~ 1 + PA.lag + NA.lag + anhedonia.lag, data = data)
summary(fit.VAR.PA)
```

```
Call:
lm(formula = PA ~ 1 + PA.lag + NA.lag + anhedonia.lag, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-22.374	-5.907	1.342	5.856	18.787

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	22.60373	6.50921	3.473	0.00107 **
PA.lag	0.39067	0.13436	2.908	0.00542 **
NA.lag	-0.12662	0.13926	-0.909	0.36759
anhedonia.lag	0.05564	0.10179	0.547	0.58711

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.769 on 50 degrees of freedom

(16 observations deleted due to missingness)

Multiple R-squared: 0.1722, Adjusted R-squared: 0.1226

F-statistic: 3.468 on 3 and 50 DF, p-value: 0.02288

```
# Linear regression model for NA
fit.VAR.NA = lm(NA. ~ 1 + PA.lag + NA.lag + anhedonia.lag, data = data)
summary(fit.VAR.NA)
```

Call:

```
lm(formula = NA. ~ 1 + PA.lag + NA.lag + anhedonia.lag, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-14.413	-5.302	-2.096	3.864	33.201

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	14.84269	6.66252	2.228	0.0304 *
PA.lag	-0.02541	0.13752	-0.185	0.8542
NA.lag	0.21894	0.14254	1.536	0.1309
anhedonia.lag	0.19856	0.10419	1.906	0.0624 .

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.999 on 50 degrees of freedom  
 (16 observations deleted due to missingness)  
 Multiple R-squared: 0.2255, Adjusted R-squared: 0.179  
 F-statistic: 4.852 on 3 and 50 DF, p-value: 0.00486

```
# Linear regression model for anhedonia
fit.VAR.anhedonia = lm(anhedonia ~ 1 + PA.lag + NA.lag + anhedonia.lag, data = data)
summary(fit.VAR.anhedonia)
```

Call:

```
lm(formula = anhedonia ~ 1 + PA.lag + NA.lag + anhedonia.lag,
    data = data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-29.750	-9.599	1.195	10.194	29.364

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	14.94851	8.92402	1.675	0.1002
PA.lag	0.04386	0.18420	0.238	0.8128
NA.lag	0.32640	0.19093	1.710	0.0936 .
anhedonia.lag	0.30771	0.13955	2.205	0.0321 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.39 on 50 degrees of freedom  
 (16 observations deleted due to missingness)  
 Multiple R-squared: 0.268, Adjusted R-squared: 0.2241  
 F-statistic: 6.102 on 3 and 50 DF, p-value: 0.001276

```
# Estimate variance-covariance matrix of the errors
res = cbind(residuals(fit.VAR.PA),residuals(fit.VAR.NA),residuals(fit.VAR.anhedonia))
cov(res)
```

	[,1]	[,2]	[,3]
[1,]	90.034929	2.375884	16.02124
[2,]	2.375884	94.326082	37.21901
[3,]	16.021244	37.219012	169.22933

## How to run the shiny app?

The shiny application is associated to a package that is stored here: [https://gitlab.kuleuven.be/ppw-okpiv/researchers/u0148925/shinyapp-paa\\_var\\_n1](https://gitlab.kuleuven.be/ppw-okpiv/researchers/u0148925/shinyapp-paa_var_n1). To install the package and run the shiny app, please use the following R code in a new script or in a R terminal:

```
# Install the package
remotes::install_gitlab("ppw-okpiv/researchers/u0148925/shinyapp-paa_var_n1", host="https:

# Import the package in the R session
library(paavar1)

# Run the shiny app
run_paa_var1()
```

## Exercise: Power analysis of VAR(1) with 3 variables

### Power analysis result

Running the simulation with the application, you should end up with a similar plot:

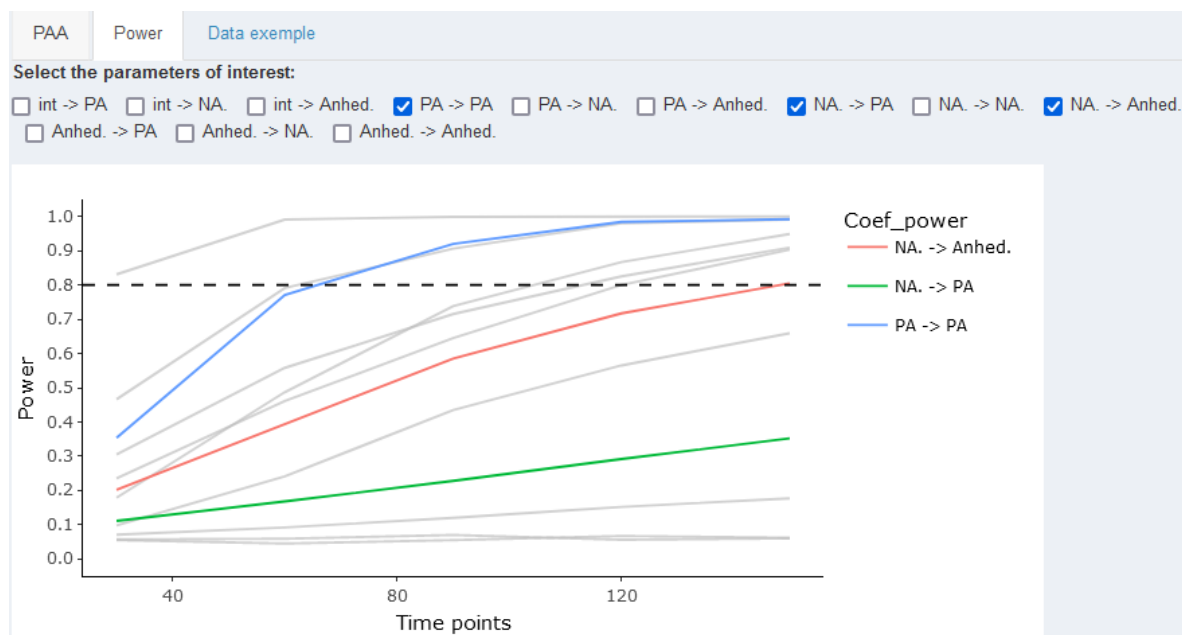


Figure 1: Power analysis VAR(1)



## Sensitivity analysis for power: varying parameters

We slightly changed the values of three coefficients to investigate how they change either the sample size recommendation or the precision of estimates:

- $\beta_{11} = .39$  to  $\beta_{11} = .8$
- $\sigma_{00} = 90$  to  $\sigma_{00} = 180$
- $R = 1000$  to  $R = 100$

What conclusions can you draw based on the following power curves?

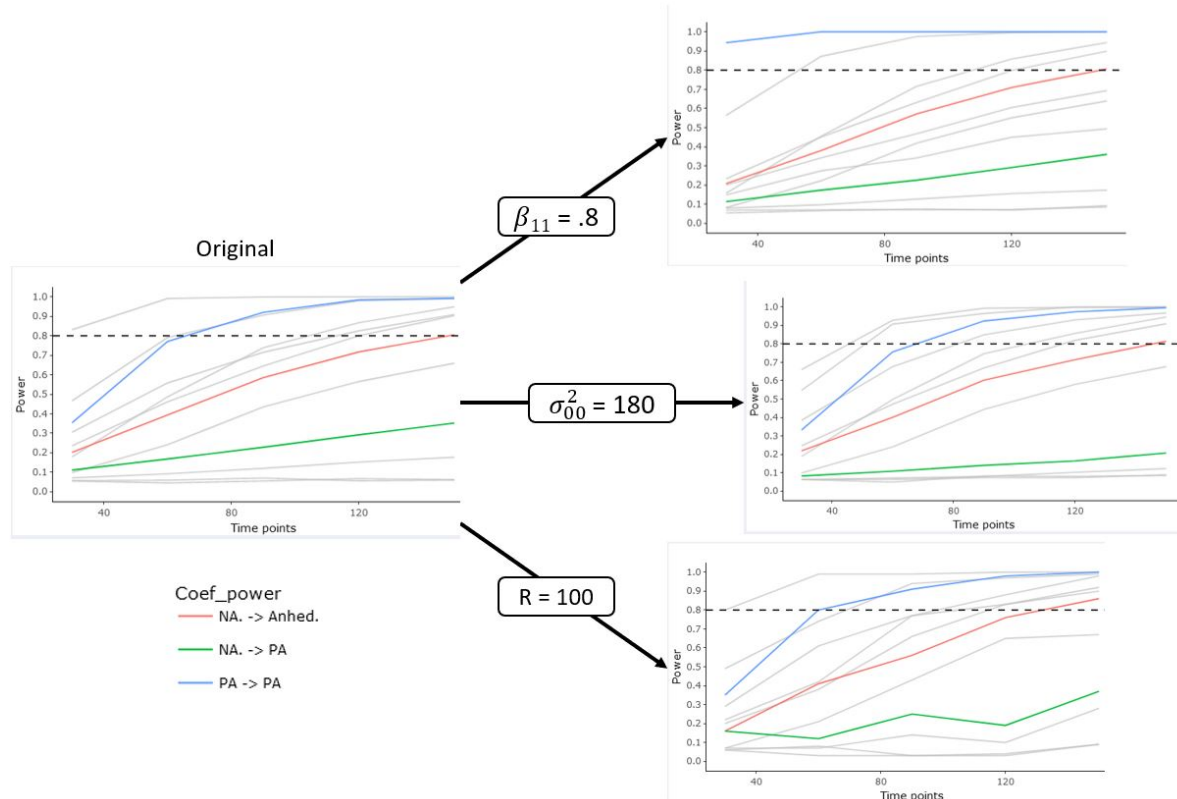


Figure 2: Sensitivity analysis for power

## Sensitivity analysis for power: using CI

Following Lafit, Revol et al. (under review), we run a sensitivity analysis using the upper and lower boundaries of the estimated coefficients of interest. First, we extract the 95% confidence interval of the estimated values of each parameter.

```
# Linear regression model for PA
confint(fit.VAR.PA, level=0.95)
```

	2.5 %	97.5 %
(Intercept)	9.5296021	35.6778667
PA.lag	0.1208077	0.6605412
NA.lag	-0.4063407	0.1530965
anhedonia.lag	-0.1488169	0.2600881

```
# Linear regression model for NA
confint(fit.VAR.NA, level=0.95)
```

	2.5 %	97.5 %
(Intercept)	1.46062438	28.2247622
PA.lag	-0.30162831	0.2508175
NA.lag	-0.06736614	0.5052475
anhedonia.lag	-0.01071022	0.4078257

```
# Linear regression model for anhedonia
confint(fit.VAR.anhedonia, level=0.95)
```

	2.5 %	97.5 %
(Intercept)	-2.97591249	32.8729333
PA.lag	-0.32611856	0.4138472
NA.lag	-0.05709339	0.7098859
anhedonia.lag	0.02741213	0.5880142

We only varied the parameter values for the auto-regressive effect of PA ( $\beta_{11}$ ) following the confidence interval. We run two new power analyses. The results are displayed below. What conclusions can you draw based on the following power curves?

### Exercise: PAA of VAR(1) with 3 variables

#### PAA result

Running the simulation with the application, you should end up with a similar plot:

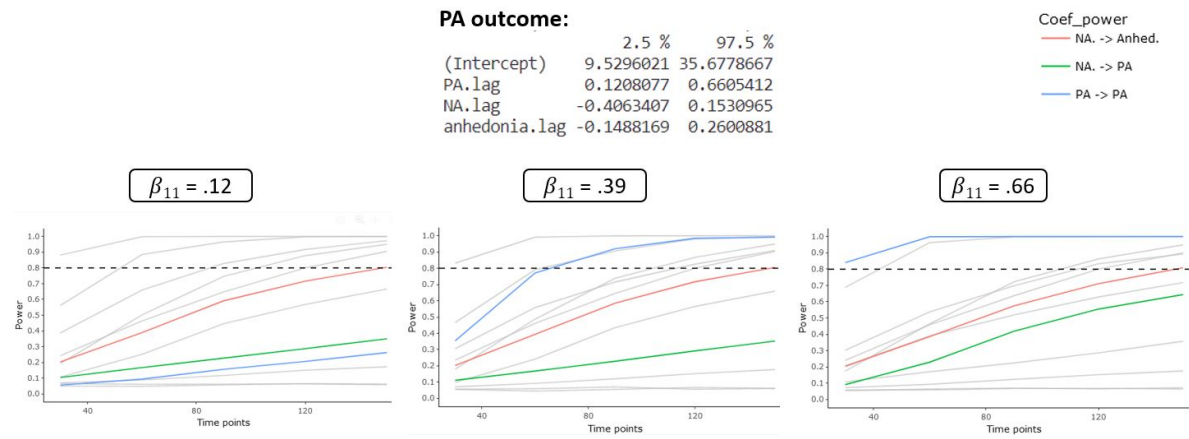


Figure 3: Sensitivity analysis for power

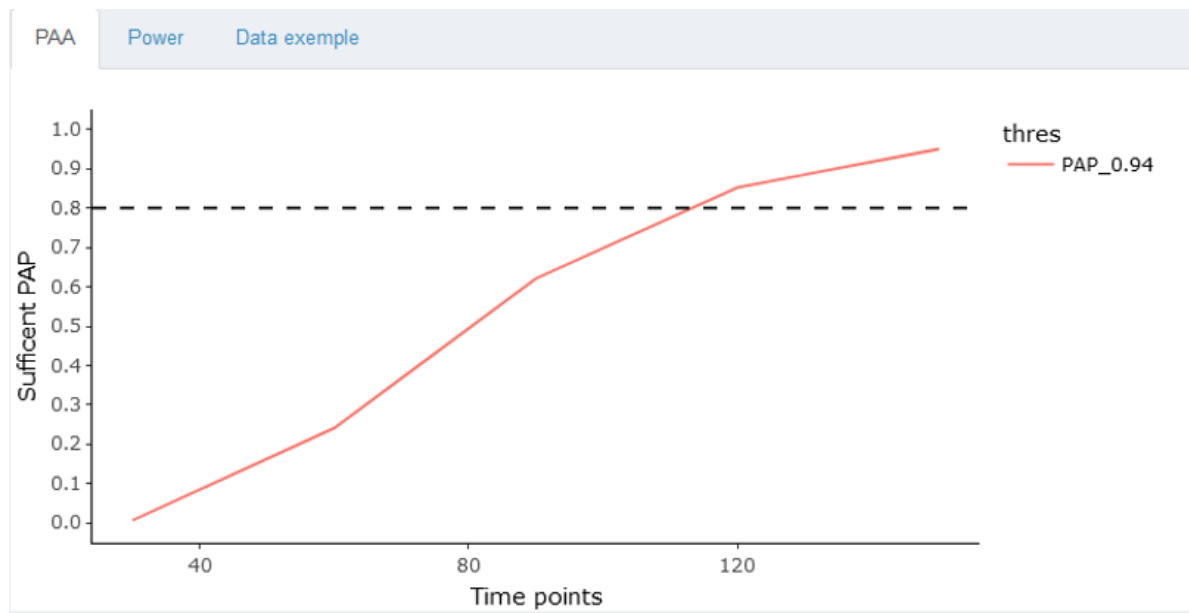


Figure 4: PAA for VAR(1)

## Sensitivity analysis for power: varying parameters

We changed the values of the transition matrix to investigate how it changes the sample size recommendation. What conclusions can you draw based on the following power curves?

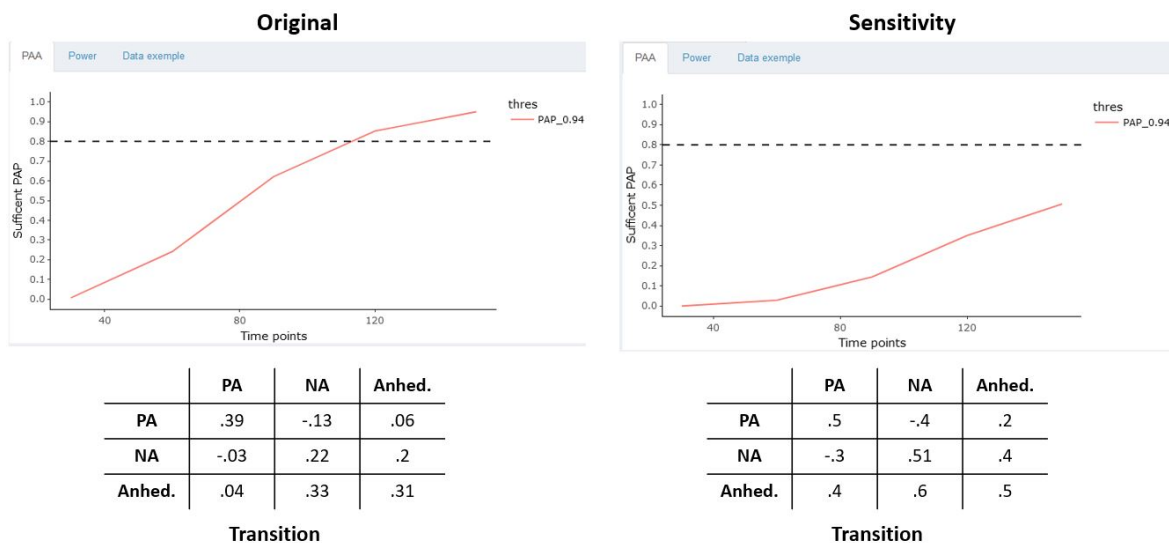


Figure 5: Sensitive PAA for VAR(1)

Note that, despite the raising of the coefficients, the new transition matrix still fulfills the stationary assumption. Higher coefficients could lead to a violation of this assumption.

## Get the session info

Below we provide the `session` information (i.e., operating system, details about the R installation, and so on) for reproducibility purposes.

```
sessionInfo()
```

```
R version 4.3.0 (2023-04-21)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Ventura 13.3.1
```

```
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] MASS_7.3-58.4      lubridate_1.9.2    forcats_1.0.0      stringr_1.5.0
[5] dplyr_1.1.2        purrr_1.0.1        readr_2.1.4        tidyr_1.3.0
[9] tibble_3.2.1       tidyverse_2.0.0    ggplot2_3.4.2      psych_2.3.3
[13] data.table_1.14.8
```

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

```
[1] gtable_0.3.3      jsonlite_1.8.4     compiler_4.3.0     tidyselect_1.2.0
[5] parallel_4.3.0    scales_1.2.1       yaml_2.3.7         fastmap_1.1.1
[9] lattice_0.21-8    R6_2.5.1           labeling_0.4.2     generics_0.1.3
[13] knitr_1.43        munsell_0.5.0      pillar_1.9.0       tzdb_0.4.0
[17] rlang_1.1.1       utf8_1.2.3         stringi_1.7.12     xfun_0.39
[21] timechange_0.2.0  cli_3.6.1          withr_2.5.0        magrittr_2.0.3
[25] digest_0.6.31     grid_4.3.0         rstudioapi_0.14    hms_1.1.3
[29] lifecycle_1.0.3   nlme_3.1-162       vctrs_0.6.2        mnormt_2.1.1
[33] evaluate_0.21     glue_1.6.2         farver_2.1.1       fansi_1.0.4
[37] colorspace_2.1-0  rmarkdown_2.22     tools_4.3.0        pkgconfig_2.0.3
[41] htmltools_0.5.5
```