

La modélisation par équations structurelles – Un guide d'accompagnement pour l'interface R

Karianne Dion ^a , Alexandra Maria Bodnaruc ^b , Geneviève Trudel ^a , Jani Lamarche ^{*a} , Valérie Ranger ^{*a} , Sophie Fobert ^{*a} , Kinsey Antonina Church ^{*a} , Joana Ntumba Mukunzi ^{*a} & Jean-Louis René ^{*a}

^aÉcole de psychologie, Université d'Ottawa

^bÉcole des sciences de l'activité physique, Université d'Ottawa

Abstract ■ La modélisation par équations structurelles avec MPlus (Caron, 2019) est un ouvrage de référence qui, divisé en trois parties, aborde les fonctions de base de Mplus (Partie 1 – Les rudiments), le traitement de données avec Mplus (Partie 2 – Traitement de données), ainsi que l'exécution d'un ensemble d'analyses statistiques impliquant de la modélisation par équations structurelles en utilisant Mplus (Partie 3 – Les analyses). Reconnaisant l'utilisation croissante de la modélisation par équations structurelles dans le domaine des sciences sociales, ainsi que le nombre limité de ressources éducatives disponibles à ce sujet, nous proposons un guide d'accompagnement pour la troisième partie du livre de Caron (2019), qui étendra son application à l'interface de programmation R. R est une interface de programmation libre d'accès et très versatile qui, similairement à Mplus, permet la réalisation d'analyses statistiques impliquant de la modélisation par équations structurelles. L'objectif de cet article est de présenter la traduction en langage de programmation R des syntaxes de la troisième partie du livre de Caron (2019) ainsi que les résultats des sorties associées. Puisque cet article vise à servir de guide complémentaire au livre de Caron (2019), les bases théoriques sous-tendant les analyses statistiques qui y sont couvertes ainsi que l'interprétation des résultats issus de celles-ci ne font pas l'objet du présent article. Cet article couvre les analyses suivantes : la régression logistique, l'analyse de trajectoire, l'analyse factorielle exploratoire et confirmatoire, la médiation, la modulation, la médiation modérée, l'analyse de classes latentes, l'analyse de modèles autorégressifs et autorégressifs croisés, l'analyse de trajectoire latente, et l'analyse de groupes multiples.

Keywords ■ modélisation par équations structurelles, langage R, régression, analyse de trajectoire, analyse factorielle, médiation, modulation, analyse de classes latentes, analyse de modèles autorégressifs, analyse de groupes multiples. **Tools** ■ R.

✉ kdion066@uottawa.ca

[10.20982/tqmp.17.3.p198](https://doi.org/10.20982/tqmp.17.3.p198)

Acting Editor ■ Denis Cousineau (Université d'Ottawa)

Reviewers

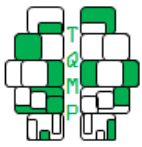
■ Two anonymous reviewers.

**: Indique une contribution égale. L'ordre de ces auteurs a été déterminé aléatoirement.*

Introduction

La modélisation par équations structurelles est une approche statistique de plus en plus répandue dans le domaine des sciences sociales. Toutefois, un nombre relativement limité de ressources existent pour guider les étudiants, professeurs et chercheurs dans la réalisation de ce type d'analyses statistiques. Pour répondre à ce besoin, CARON (2019) a publié le livre « La modélisation par équations structurelles avec Mplus », un guide d'introduc-

tion à la modélisation par équations structurelles à l'aide du logiciel statistique Mplus (MUTHÉN & MUTHÉN, 1998). Le livre est divisé en trois parties majeures, la première partie expliquant les bases de l'utilisation du logiciel Mplus, la seconde (Partie 2 - Traitement des données) portant sur le traitement des données avec Mplus et, finalement, la troisième (Partie 3 - Les analyses) décrivant les étapes d'exécution de différents types d'analyses statistiques employant la modélisation par équations structurelles en utilisant Mplus. Bien que cet ouvrage soit une ressource très



pratique, son utilisation est limitée à un seul logiciel. Outre Mplus, l'interface de programmation R (REVELLE, 2020) constitue une option fort intéressante pour la réalisation d'analyses statistiques impliquant de la modélisation par équations structurelles. Étant libre d'accès et très versatile, l'interface R est d'ailleurs de plus en plus répandue.

L'objectif du présent article est de traduire en langage de programmation R les syntaxes de la troisième partie du livre de CARON (2019) ainsi que de présenter les résultats des sorties associées. Cet article vise à servir de guide complémentaire au livre de CARON (2019). Ainsi, les bases théoriques sous-tendant les analyses statistiques qui sont présentées dans les sections ci-dessous et l'interprétation des résultats issus de celles-ci ne font pas l'objet du présent article. L'article couvre les analyses suivantes : la régression logistique, l'analyse de trajectoire, l'analyse factorielle exploratoire et confirmatoire, la médiation, la modération, la médiation modérée, l'analyse de classes latentes, l'analyse de modèles autorégressifs et autorégressifs croisés, l'analyse de trajectoire latente, et l'analyse de groupes multiples. Bien que la majorité des syntaxes présentées dans cet article soient équivalentes à celles du livre de Caron, il est important de mentionner que certaines commandes effectuées dans Mplus ne peuvent être traduites en langage R. Lorsque la traduction exacte n'est pas possible, les syntaxes présentées sont celles qui s'apparentent le plus aux syntaxes du livre de CARON (2019).

Instructions d'utilisation

Dans le présent ouvrage, la structure des syntaxes et des sorties R pour chaque analyse en annexe est organisée de manière similaire au livre de CARON (2019). De règle générale, chacune des analyses statistiques mentionnées plus haut est illustrée à l'aide d'un exemple comprenant une syntaxe et une sortie. Les sections suivantes expliquent comment lire et utiliser ces syntaxes et ces sorties de manière pratique.

La syntaxe

Chaque syntaxe est précédée d'une légende (intitulée *Listing*) qui définit le type d'analyse présenté, le fichier de données utilisé, les variables sélectionnées, les librairies nécessaires ainsi que le fichier de syntaxe R correspondant. Toutes les syntaxes présentées suivent sensiblement les mêmes étapes et chacune de ses étapes sont clairement identifiées dans la syntaxe par un signe de dièse (#). Les commandes à effectuer sont présentées directement à la suite des étapes identifiées. De cette façon, les commandes de syntaxes peuvent être copiées dans R et modifiées (p. ex., noms du répertoire de travail et de la base de données) afin de conduire l'analyse voulue.

Les étapes 1 à 7 pour le traitement et l'analyse des données pourraient, entièrement ou en partie, être applicables à chacune des syntaxes présentées en annexe :

1. Définir le répertoire de travail. Le répertoire de travail est l'endroit sur le disque dur où sont sauvegardées les bases de données et autres fichiers utilisés pour effectuer les analyses statistiques. Dans R, le répertoire de travail est défini à l'aide de la commande

```
setwd("chemin du répertoire désiré").
```

Une fois le répertoire de travail défini via cette commande, il n'est plus essentiel de spécifier cette information dans les commandes subséquentes d'importation et d'exportation de fichiers.

2. Installer et sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse. La majorité des analyses présentées dans ce manuel nécessite de télécharger au moins une librairie R (p.ex., lavaan, psych). La commande

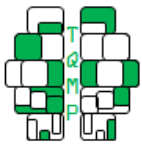
```
install.packages("nom de librairie",  
dependencies = TRUE)
```

permet d'installer une librairie ne faisant pas déjà partie des librairies de base de R. Une fois installée, la librairie doit être activée à l'aide de la commande

```
library(nom de librairie)
```

Les instructions concernant le téléchargement des librairies se retrouvent également dans le manuel *An Introduction to R* (VENABLES, SMITH & R CORE TEAM, 2020).

3. Importer la base de données. Les bases de données accompagnant le livre de CARON (2019) sont fournies sous le format .dat et sont disponibles en ligne à l'adresse suivante : <https://www.puq.ca/catalogue/livres/modelisation-par-equations-structurelles-avec-mplus-3563.html>. Puisque les fichiers de type .dat sont moins bien supportés par R, les bases de données ont été converties au format .csv pour l'ensemble des syntaxes présentées dans ce manuel. Préalablement, le nom des variables a été ajouté dans la première rangée de chaque base de données et la colonne « ID » a été supprimé des bases de données dans lesquelles elle était incluse. Alors que l'importation des fichiers .csv est supportée par les librairies de base de R, celle de plusieurs autres formats (p.ex., .xlsx, .sav, .dta) de fichiers fréquemment utilisés requiert l'installation de librairies additionnelles (p.ex., xlsx, foreign). Les manuels *An Introduction to R* (VENABLES et collègues., 2020) et *Data Import / Export* (R CORE TEAM, 2020b) fournissent davantage d'information sur l'importation et l'exportation de fichiers sous différents formats.
4. Remplacer les valeurs manquantes codées 999 par NA. Les valeurs manquantes, initialement codées « 999 »



(notation standard pour les données manquantes de Mplus) ont été remplacées par « NA » (notation standard pour les données manquantes de R) à l'aide de la commande

```
data[data == 999] <- NA
```

où `data` correspond au nom de la base de données.

5. Inspecter la base de données. Les commandes `str(data)` et `md.pattern(data)`, où « `data` » correspond au nom de la base de données, ont été utilisées pour visualiser un aperçu de la base de données et, lorsque applicable, du patron de données manquantes.
6. Analyses descriptives (i.e., moyennes, matrices de covariance, matrices de corrélations). Lorsque nécessaire, les commandes `summary()`, `cov()` et `cor()` ont été utilisées pour obtenir les mesures de tendance centrale ainsi que les matrices de covariance et de corrélation.
7. Modifier la base de données. Lorsque nécessaire, le type de variable (p.ex., continue, ordinale, dichotomique, etc.) a été spécifié et/ou de nouvelles variables ont été créées à l'aide de commandes appropriées.

Les étapes 8 à 10 varient selon l'analyse abordée.

8. Définir le modèle.
9. Spécifier les paramètres du modèle.
10. Obtenir le sommaire des estimés du modèle.

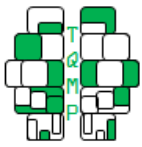
La sortie

Chaque syntaxe est suivie d'un exemple de sortie. Les étapes à suivre pour arriver aux sorties sont clairement identifiées à l'aide de dièses et sont succédées par les commandes à effectuer dans R. Similairement aux syntaxes, les commandes de sorties peuvent être copiées et modifiées au besoin. Comme dans l'ouvrage de CARON (2019), un code de couleurs a été utilisé dans les sorties pour mettre en valeur les informations importantes. La légende (intitulée *Output*) précédant la sortie définit ce que représente les différentes couleurs pour chacune des analyses. Lorsque possible, ce code de couleur s'apparente à celui utilisé par CARON (2019). L'interprétation de la sortie et la rédaction des résultats sont décrites en détails dans le livre de CARON (2019). Les sorties du présent ouvrage ont été obtenues avec la version 4.0.1 de R (R CORE TEAM, 2020a) qui peut être téléchargée à l'adresse suivante : <http://www.r-project.org/>.

Les analyses

Les analyses présentées correspondent aux analyses de modélisation par équations structurelles présentées dans le livre de CARON (2019). Voici la liste complète des types d'analyses qui se retrouvent en annexe :

- La régression logistique. Le Listing 1 présente les commandes R nécessaires pour effectuer une régression logistique via la fonction `glm` de la librairie de base `stats`. La librairie `mice` (van BUUREN & GROOTHUIS-ODSHOORN, 2011; van BURREN et collègues., 2020) a été utilisée pour examiner le patron de données manquantes. La sortie de l'analyse se retrouve à l'Output 1. À titre de référence, cette analyse correspond au Chapitre 9 (pp. 18-86) du livre de CARON (2019).
- L'analyse de trajectoire. Le Listing 2 présente les commandes R nécessaires pour compléter une analyse de trajectoire via la fonction `sem` de la librairie `lavaan` (ROSSEEL, 2012; ROSSEEL et collègues., 2020). La librairie `mice` (van BUUREN & GROOTHUIS-ODSHOORN, 2011; van BURREN et collègues., 2020) a été utilisée pour inspecter le patron de données manquantes. La sortie qui en découle est présentée à l'Output 2. Cette analyse correspond au Chapitre 10 (pp. 87-93) du livre de CARON (2019).
- L'analyse factorielle exploratoire. Trois modèles d'analyse factorielle exploratoire sont présentés. Le modèle 1 (voir Listing 3.1 et Output 3.1) correspond à une analyse à un seul facteur, le modèle 2 (voir Listing 3.2 et Output 3.2) contient deux facteurs puis le modèle 3 comprend trois facteurs (voir Listing 3.3 et Output 3.3). L'analyse est effectuée via la fonction `fa` de la librairie `psych` (REVELLE, 2020). La librairie `GPArotation` a été utilisée pour spécifier le type de rotation des modèles testés. Cette analyse correspond au Chapitre 11 (pp. 95-109) du livre de CARON (2019).
- L'analyse factorielle confirmatoire. Deux modèles d'analyse factorielle confirmatoire sont présentés, un avec des items de référence par défaut (voir Listing 4.1 et Output 4.1) et l'autre avec des items de référence spécifiés (Listing 4.2 et Output 4.2). Les analyses sont effectuées via la fonction `cfa` de la librairie `lavaan` (ROSSEEL, 2012; ROSSEEL et collègues., 2020). Cette analyse correspond au Chapitre 12 (pp. 112-122) du livre de CARON (2019).
- L'analyse de médiation. L'analyse de médiation est présentée via la fonction `sem` de la librairie `lavaan` (ROSSEEL, 2012; ROSSEEL et collègues., 2020). Les librairies `boot` (DAVISON & HINKLEY, 1997; CANTY & RIPLEY, 2020) et `mice` (van BUUREN & GROOTHUIS-ODSHOORN, 2011; van BURREN et collègues., 2020) ont également été utilisées. L'analyse est présentée pour un modèle sans ré-échantillonnage (voir Listing 5.1 et Output 5.1) et avec ré-échantillonnage (voir Listing 5.2 et Output 5.2). Cette analyse correspond au Chapitre 13 (pp. 123-139) du livre de CARON (2019).
- L'analyse de modération. L'analyse de modération est présentée de deux manières, une via la fonction `lm`



(voir Listing 6.1 et Output 6.1) de la librairie `gvlma` (PEÑA & SLATE, 2006, 2019) et l'autre via la fonction `sem` (voir Listing 6.2 et Output 6.2). La librairie `mice` (van BUUREN & GROOTHUIS-ODSHOORN, 2011; van BURREN et collègues., 2020) a été utilisée pour inspecter le patron de données manquantes. Cette analyse correspond au Chapitre 14 (pp. 140-147) du livre de CARON (2019).

- L'analyse de médiation modérée. L'analyse de médiation modérée est démontrée via la fonction `sem`. Deux modèles de l'analyse sont présentés, un avec un modérateur dichotomique (voir Listing 7.1 et Output 7.1) et l'autre avec un modérateur continu (voir Listing 7.2 et Output 7.2). Cette analyse correspond au Chapitre 15 (pp. 149-163) du livre de CARON (2019).
- L'analyse de classes latentes. Puisqu'il n'existe présentement aucune librairie R permettant d'effectuer la procédure en trois étapes avec des variables continues présentée par CARON (2019), l'analyse de classes latentes est présentée de manière différente. Premièrement, l'analyse est présentée avec la librairie `depmixS4` (VISSER & SPEEKENBRINK, 2010, 2020) avec deux (voir Listing 8.1.1 et Output 8.1.1), trois (voir Listing 8.1.2 et Output 8.1.2), quatre (voir Listing 8.1.3 et Output 8.1.3) et cinq (voir Listing 8.1.4 et Output 8.1.4) classes latentes. La librairie `depmixS4` permet d'obtenir les indices d'ajustement pour des modèles dont le nombre de classes latentes est spécifié dans la commande. Toutefois, cette librairie n'inclut pas l'entropie comme mesure d'ajustement et ne permet pas d'effectuer la procédure en trois étapes. Deuxièmement, l'analyse de classes latentes est présentée avec la librairie `mclust` (SCRUCCA, FOP, MURPHY & RAFTERY, 2016; FRALEY, RAFTERY, SCRUCCA, MURPHY & FOP, 2020) (voir Listing 8.2 et Output 8.2). Cette librairie estime automatiquement la structure du meilleur modèle et fournit les indices d'ajustement associés. Cependant, les librairies `depmixS4` et `mclust` n'incluent pas l'entropie comme mesure d'ajustement et ne permet pas la procédure en trois étapes. L'analyse de classes latentes correspond au Chapitre 16 (pp. 165-184) du livre de CARON (2019).
- L'analyse de modèles autorégressifs. Le Listing 9 présente les commandes R nécessaires pour réaliser une analyse de modèles autorégressifs via la fonction `sem`. La sortie qui en découle est présentée à l'Output 9. Cette analyse correspond au Chapitre 17 (pp. 185-192) du livre de CARON (2019).
- L'analyse de modèles autorégressifs croisés. La syntaxe R pour réaliser une analyse de modèles autorégressifs croisés via la fonction `sem` est présentée au Listing 10. La sortie de l'analyse se retrouve à l'Output 10. L'analyse de modèles autorégressifs croisés correspond au

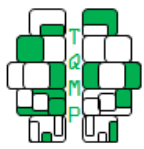
Chapitre 18 (pp. 193-199) du livre de CARON (2019).

- L'analyse de trajectoire latente. L'analyse de trajectoire latente est présentée via la fonction `growth` de la librairie `lavaan` (ROSSEEL, 2012; ROSSEEL et collègues., 2020). Le modèle 1 présente l'analyse avec un modèle linéaire sans covariable (voir Listing 11.1 et Output 11.1), le modèle 2 présente l'analyse avec un modèle quadratique sans covariable (voir Listing 11.2 et Output 11.2), le modèle 3 présente l'analyse linéaire avec une covariable stable temporellement (voir Listing 11.3 et Output 11.3), le modèle 4 présente l'analyse linéaire avec une covariable instable temporellement (voir Listing 11.4 et Output 11.4) puis le modèle 5 présente l'analyse linéaire avec des trajectoires latentes parallèles (voir Listing 11.5 et Output 11.5). Cette section correspond au Chapitre 19 (pp. 201-247) du livre de CARON (2019).
- L'analyse de groupes multiples. L'analyse de groupe multiple est présentée via la fonction `sem` et correspond au modèle à trois étapes présentées au Chapitre 20 (pp. 249-256) de CARON (2019) : l'étape 0 étant le modèle sans contrainte (voir Listing 12.1 et Output 12.1), l'étape 1 étant les coefficients de régression (voir Listing 12.2 et Output 12.2), l'étape 2 étant les ordonnées à l'origine (voir Listing 12.3 et Output 12.3) puis l'étape 3 étant les covariances résiduelles (voir Listing 12.4 et Output 12.4).

Discussion

La modélisation par équations structurelles devient de plus en plus populaire auprès des étudiants, des chercheurs et des professionnels en sciences sociales. Toutefois, le nombre de ressources disponibles pour ce type d'analyse est très limité, surtout en français. Pour répondre à ce besoin, CARON (2019) a créé un livre pratique détaillant 12 analyses de modélisation par équations structurelles avec le logiciel Mplus. L'auteur offre une définition de l'analyse, la syntaxe et la sortie Mplus ainsi qu'une interprétation des résultats. Le présent article avait comme objectif d'étendre la portée des travaux de CARON (2019) en traduisant les commandes Mplus en langage R pour chacune des analyses présentées. Comme il s'agit de deux logiciels différents, certaines commandes Mplus n'ont pas pu être converti directement en langage R. Lorsque cette limitation est survenue, des commandes R équivalentes ou similaires ont été offertes.

Cet article s'ajoute à la courte liste des ressources francophones disponibles sur la modélisation par équations structurelles tout en rendant cette information accessible en fournissant une syntaxe R simple et pratique. Les commandes fournies sont aussi facilement modifiables pour être compatible avec la base de données voulue. Le livre

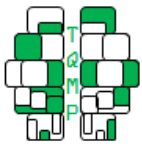


de CARON (2019) représente une excellente ressource afin d'obtenir plus d'information sur les analyses effectuées. Prochainement, il pourrait être bénéfique d'étendre ce type de travaux à d'autres logiciels statistiques et à d'autres types d'analyses de modélisation par équation structurelles plus complexes.

Références

- CANTY, A. & RIPLEY, B. D. (2020). Package 'boot' (Version 1.117). Récupérée, à partir de <https://cran.r-project.org/package=boot>
- CARON, P.-O. (2019). *La modélisation par équations structurelles avec Mplus*. Montréal : Presses de l'Université du Québec.
- DAVISON, A. C. & HINKLEY, D. V. (1997). *Bootstrap Methods and Their Applications*. Cambridge : Cambridge University Press.
- FRALEY, C., RAFTERY, A. E., SCRUTTA, L., MURPHY, T. B. & FOP, M. (2020). mclust : Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation. *CRAN Repository* : 1-169. Récupérée à partir de <https://cran.r-project.org/package=mclust>
- MUTHÉN, L. & MUTHÉN, B. (1998). *Mplus user's guide* (7e éd.). É. Los Angeles : StatModels.
- PEÑA, E. A. & SLATE, E. H. (2006). Global validation of linear model assumptions. *Journal of the American Statistical Association*, 101(473), 341-354. Récupérée à partir de <https://doi.org/10.1198/016214505000000637>
- PEÑA, E. A. & SLATE, E. H. (2019). Package 'gvlma' (Version 1.16). Récupérée, à partir de <https://cran.r-project.org/package=gvlma>
- R CORE TEAM. (2020a). R (version 4.0.2) Data Import / Export (Version 1.31). Récupérée, à partir de <https://cran.r-project.org/doc/manuals/r-release/R-data.pdf>
- R CORE TEAM. (2020b). R Installation and Administration (Version 4.0.2). Récupérée, à partir de <https://cran.r-project.org/doc/manuals/r-release/R-admin.pdf>
- REVELLE, W. (2020). Package 'psych'. Récupérée, à partir de <https://cran.r-project.org/package=psych>
- ROSSEEL, Y. (2012). Lavaan : An R package for structural equation modeling. *Journal of Statistical Software*, 48(2), 1-36. Récupérée à partir de <https://doi.org/10.18637/jss.v048.i02>
- ROSSEEL, Y., JORGENSEN, T. D., ROCKWOOD, N., OBERSKI, D., BYRNES, J., VANBRABAN, L., ... DU, H. (2020). Package 'lavaan' (Version 1.104). Récupérée, à partir de <https://cran.r-project.org/package=lavaan>
- SCRUTTA, L., FOP, M., MURPHY, T. B. & RAFTERY, A. E. (2016). mclust 5 : clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1), 289-317. Récupérée à partir de <https://doi.org/10.32614/RJ-2016-021>
- van BURREN, S., GROOTHUIS-ODDSHOORN, K., GERKO, V., SCHOUTEN, R., ROBITZSCH, A., ROCKENSCHAUB, P., ... GRAY, V., B. and Arel-Bundock. (2020). Package 'mice'. Récupérée, à partir de <https://cran.r-project.org/package=mice>
- van BUUREN, S. & GROOTHUIS-ODDSHOORN, K. (2011). mice : Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3), 1-67. Récupérée à partir de <https://doi.org/10.18637/jss.v045.i03>
- VENABLES, W. N., SMITH, D. M. & R CORE TEAM. (2020). An Introduction to R version 4.0.2. Récupérée, à partir de <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>
- VISSEER, I. & SPEEKENBRINK, M. (2010). depmixS4 : An R-package for hidden Markov Models. *Journal of Statistical Software*, 36(7), 1-21. Récupérée à partir de <https://doi.org/10.18637/jss.v036.i07>
- VISSEER, I. & SPEEKENBRINK, M. (2020). Package 'depmixS4' (Version 1.52). Récupérée, à partir de <https://cran.r-project.org/web/packages/depmixS4/index.html>

Les annexes suivent.



Annexe 1 : La régression logistique (*Logistic regression*)

Listing 1. Syntaxe pour réaliser une régression logistique. Le fichier de données utilisé est `reg_log_data.csv`, les variables utilisées sont VI1, VI2 et VD, puis les librairies nécessaires sont `stats` et `mice`. Le fichier de syntaxe R Chapitre 9 – Régression logistique – Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner les librairies nécessaires pour l'analyse
library(mice)
library(stats)

# Importer la base de données
data <- read.csv("reg_log_data.csv", header = TRUE, sep = ",")

# Remplacer les valeurs manquantes codées 999 (MPlus) par NA (R)
data[data == 999] <- NA

# Inspecter la base de données
str(data)
md.pattern(data)

# Modifier la base de données
# Définir la variable dichotomique
data$VD <- as.factor(data$VD)

# Exclure les données manquantes
data <- data[!(is.na(data$VI1) | is.na(data$VI2) | is.na(data$VD)),]

# Définir le modèle
model <- 'VD ~ VI1 + VI2'

# Spécifier les paramètres du modèle
reglog <- glm(model, data = data, family = "binomial")

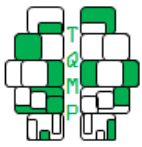
# Obtenir le sommaire des estimations du modèle
summary(reglog)

# Obtenir les rapports de chances ("odds ratios")
exp(reglog$coefficients)

# Calculer la variance expliquée (pseudo R2 de McFadden)
ll.null.model <- reglog$null.deviance / -2
ll.model <- reglog$deviance / -2
(ll.null.model - ll.model) / ll.null.model

# Calculer la valeur p associé au pseudo R2 de McFadden
1 - pchisq(2 * (ll.model - ll.null.model), df = (length
  (reglog$coefficients) - 1))
```

Output 1. Sortie de l'analyse de régression logistique du Listing 1. Les nombres en gris sont le total des observations manquantes, les nombres en rouge sont les observations avec des données manquantes pour une variable ou plus,



les nombres en **vert** sont les coefficients de régression et leur valeur p , les nombres en **jaune** sont les rapports des chances puis les nombres en **bleu** sont le pseudo R de McFadden et sa valeur p .

```
> # Inspecter la base de données
> str(data)

'data.frame': 1375 obs. of 3 variables:
 $ VI1: int  1 0 0 0 2 1 1 1 1 1 ...
 $ VI2: int  13 30 13 5 12 13 10 6 5 4 ...
 $ VD : Factor w/ 3 levels "0","1","999": 2 2 2 2 1 1 1 1 1 1 ...
```

```
> md.pattern(data)
```

	VI1	VI2	VD	
1375	1	1	1	0
9	1	1	0	1
3	1	0	1	1
158	1	0	0	2
6	0	1	1	1
6	161	167	334	

```
> # Obtenir le sommaire des estimations du modèle
> summary(reglog, fit.measure = TRUE)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.5463	-0.4160	-0.2862	-0.2112	2.8694

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-4.2546	0.2199	-19.351	< 2e-16 ***
VI1	0.4435	0.1082	4.099	4.15e-05 ***
VI2	0.1542	0.0122	12.637	< 2e-16 ***

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

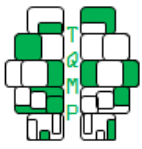
(Dispersion parameter for binomial family taken to be 1)

Null deviance: 968.49 on 1374 degrees of freedom
Residual deviance: 741.89 on 1372 degrees of freedom
AIC: 747.89

Number of Fisher Scoring iterations: 6

```
> # Obtenir les rapports de chances ("odds ratios")
> exp(reglog$coefficients)
```

(Intercept)	VI1	VI2
0.0141994	1.5582332	1.1667626



```
> # Calculer la variance expliquée (pseudo R2 de McFadden)
> (ll.null-ll.proposed) / ll.null

0.2339726

> # Calculer la valeur p associée au pseudo R2 de McFadden
> 1-pchisq(2*(ll.model-ll.null.model), df=(length(reglog$coefficients)-1))

0
```

Annexe 2 : L'analyse de trajectoire (*Path analysis*)

Listing 2. Syntaxe pour réaliser une analyse de trajectoire. Le fichier de données utilisé est `path_data.csv`, les variables utilisées sont V1 à V6, puis les librairies nécessaires sont `lavaan` et `mice`. Le fichier de syntaxe R Chapitre 10 - Analyse de trajectoire - `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)
library(mice)

# Importer la base de données
data <- read.csv("path_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)
md.pattern(data)

# Remplacer les valeurs manquantes codées 999 (MPlus) par NA (R)
data[data == 999] <- NA

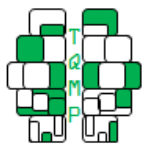
# Définir le modèle
model <- 'V6 ~ V3 + V4 + V5
         V4 ~ V1 + V2
         V3 ~ V1 + V2'

# Spécifier les paramètres du modèle
fit <- sem(model, data = data, missing = "pairwise", ordered = c("V6"))

# Obtenir le sommaire des estimations du modèle
summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 2. Sortie de l'analyse de trajectoire du Listing 2. Les nombres en **gris** sont le total des observations manquantes, les nombres en **rouge** sont les observations avec des données manquantes pour une variable ou plus, les nombres en **jaune** sont les indices d'ajustements, les nombres en **vert** sont les coefficients de régression et leur valeur *p* puis les nombres en **magenta** représentent la variance expliquée par chacune des variables dépendantes.

```
> # Inspecter la base de données
> str(data)
```

```
'data.frame': 188 obs. of 6 variables:
 $ V1: int  1 11 12 4 0 1 9 8 10 4 ...
 $ V2: int  1 3 6 2 5 3 8 6 5 0 ...
 $ V3: int  NA 20 20 19 18 18 18 17 17 17 ...
 $ V4: int  NA 70 68 17 53 50 NA 86 56 11 ...
 $ V5: int  0 9 9 12 5 6 5 11 7 6 ...
 $ V6: int  NA 1 1 1 0 1 NA 1 1 0 ...
```

```
> md.pattern(data)
```

```
      V5 V1 V2 V6 V4 V3
69      1  1  1  1  1  0
30      1  1  1  1  1  1
11      1  1  1  0  0  1
75      1  1  1  0  0  0
1       1  0  0  1  1  1
1       1  0  0  1  1  0
1       0  0  0  1  0  1
```

```
1  3  3  86  87 106 286
```

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

```
lavaan 0.6-7 ended normally after 47 iterations
```

Estimator	DWLS	
Optimization method	NLMINB	
Number of free parameters	12	
	Used	Total
Number of observations	110	188
Number of missing patterns	3	

```
Model Test User Model:
```

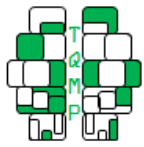
	Standard	Robust
Test Statistic	3.363	3.871
Degrees of freedom	5	5
P-value (Chi-square)	0.644	0.568
Scaling correction factor		0.912
Shift parameter		0.182
simple second-order correction		

```
Model Test Baseline Model:
```

Test statistic	17.444	16.753
Degrees of freedom	3	3
P-value	0.001	0.001
Scaling correction factor		1.050

```
User Model versus Baseline Model:
```

Comparative Fit Index (CFI)	1.000	1.000
Tucker-Lewis Index (TLI)	1.068	1.049



Robust Comparative Fit Index (CFI)	NA
Robust Tucker-Lewis Index (TLI)	NA

Root Mean Square Error of Approximation:

RMSEA	0.000	0.000
90 Percent confidence interval - lower	0.000	0.000
90 Percent confidence interval - upper	0.108	0.117
P-value RMSEA ≤ 0.05	0.762	0.700

Robust RMSEA	NA
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	NA

Standardized Root Mean Square Residual:

SRMR	0.053	0.053
------	-------	-------

Regressions:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
V6 ~						
V3	0.129	0.024	5.441	0.000	0.129	0.622
V4	0.014	0.007	2.020	0.043	0.014	0.230
V5	0.108	0.054	2.009	0.045	0.108	0.278
V4 ~						
V1	0.845	0.513	1.647	0.100	0.845	0.175
V2	2.675	0.938	2.851	0.004	2.675	0.343
V3 ~						
V1	0.417	0.183	2.281	0.023	0.417	0.288
V2	0.393	0.292	1.347	0.178	0.393	0.168

Intercepts:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.V6	0.000				0.000	0.000
.V4	27.212	5.624	4.838	0.000	27.212	1.441
.V3	5.745	1.937	2.967	0.003	5.745	1.014

Thresholds:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
V6 t1	3.279	0.522	6.284	0.000	3.279	2.786

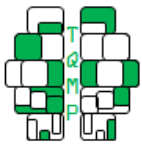
Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.V6	0.487				0.487	0.351
.V4	285.680	45.765	6.242	0.000	285.680	0.801
.V3	27.208	6.863	3.965	0.000	27.208	0.848

Scales y*:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
V6	1.000				1.000	1.000

R-Square:



	Estimate
V6	0.649
V4	0.199
V3	0.152

Annexe 3 : L'analyse factorielle exploratoire (*Exploratory factor analysis*)

Listing 3.1. Syntaxe pour réaliser une analyse factorielle exploratoire avec un modèle à un facteur (modèle 1). Le fichier de données utilisé est `efa_data.csv`, les variables utilisées sont I1 à I10, puis les librairies nécessaires sont `psych` et `GPArotation`. Le fichier de syntaxe R Chapitre 11 - Analyse factorielle exploratoire - Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaires pour l'analyse
library(psych)
library(GPArotation)

# Importer la base de données
data <- read.csv("efa_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Déterminer les valeurs propres
eigen(cor(data))

# Définir le modèle
# Modèle à 1 facteur
model.1 <- fa(data, nfactors = 1, rotate = "geominQ", fm = "ml")

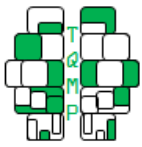
# Obtenir le sommaire des estimations du modèle
model.1

# Obtenir le CFI
((model.1$null.chisq - model.1$null.dof) - (model.1$STATISTIC - model.1$dof))
/ (model.1$null.chisq - model.1$null.dof)
```

Output 3.1. Sortie de l'analyse factorielle exploratoire du Listing 3.1. Les nombres en **magenta** sont les valeurs propres ou *eigenvalues* (une fois divisées par le nombre d'items, ces valeurs représentent la variance expliquée par chacun des facteurs), les nombres en **vert** sont les coefficients de saturation après rotation puis les nombres en **jaune** sont les indices d'ajustement.

```
> # Inspecter la base de données
> str(data) ''

data.frame: 200 obs. of 10 variables:
 $ I1 : int  9 10 6 9 12 3 5 3 14 -7 ...
 $ I2 : int  8 10 6 8 11 4 4 1 15 -9 ...
 $ I3 : int  5 6 9 6 6 3 3 4 11 -9 ...
 $ I4 : int  8 -1 3 9 9 6 6 3 12 -1 ...
 $ I5 : int  6 10 2 5 8 5 4 10 15 -8 ...
 $ I6 : int  6 2 1 14 -3 -1 6 2 5 5 ...
```



```

$ I7 : int  3 2 2 14 -2 -1 5 -1 5 8 ...
$ I8 : int  9 4 4 14 -3 1 5 1 4 5 ...
$ I9 : int  7 1 2 13 -3 -1 5 5 4 10 ...
$ I10 : int -1 6 2 15 3 -2 2 4 4 1 ...

> # Déterminer les valeurs propres
> eigen(cor(data))

[1] 3.78346430 3.31600601 0.71192274 0.66380763 0.43375651
[6] 0.39282121 0.27751486 0.24668012 0.09628986 0.07773674

> # Obtenir le sommaire des estimations du modèle
> model.1

Factor Analysis using method = ml
Call: fa(r = data, nfactors = 1, rotate = "geominQ", fm = "ml")

Standardized loadings (pattern matrix) based upon correlation matrix

```

	ML1	h2	u2	com
I1	-0.06	0.0038	1.00	1
I2	-0.08	0.0059	0.99	1
I3	-0.06	0.0039	1.00	1
I4	-0.13	0.0160	0.98	1
I5	-0.15	0.0234	0.98	1
I6	0.99	0.9704	0.03	1
I7	0.91	0.8365	0.16	1
I8	0.79	0.6186	0.38	1
I9	0.76	0.5713	0.43	1
I10	0.52	0.2669	0.73	1

```

          ML1
SS loadings  3.32
Proportion Var 0.33

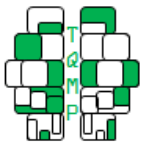
Mean item complexity = 1
Test of the hypothesis that 1 factor is sufficient.

The degrees of freedom for the null model are 45 and the objective function was
7.57 with Chi Square of 1474.14
The degrees of freedom for the model are 35 and the objective function was 3.8

The root mean square of the residuals (RMSR) is 0.3
The df corrected root mean square of the residuals is 0.34

The harmonic number of observations is 200 with the empirical chi square 1624.76
with prob < 1.899979e-319
The total number of observations was 200 with Likelihood Chi Square = 738.72
with prob < 1.1e-132

```



Tucker Lewis Index of factoring reliability = 0.365 RMSEA index = 0.317 and the 90
 % confidence intervals are 0.298 0.338

BIC = 553.27

Fit based upon off diagonal values = 0.52

Measures of factor score adequacy

	ML1
Correlation of (regression) scores with factors	0.99
Multiple R square of scores with factors	0.98
Minimum correlation of possible factor scores	0.95

```
> # Obtenir le CFI
> ((model.1$null.chisq - model.1$null.dof) - (model.1$STATISTIC - model.1$dof))
  / (model.1$null.chisq - model.1$null.dof)
```

```
[1] 0.5075971
```

Listing 3.2. Syntaxe pour réaliser une analyse factorielle exploratoire avec un modèle à deux facteurs (modèle 2). Le fichier de données utilisé est `efa_data.csv`, les variables utilisées sont I1 à I10, puis les librairies nécessaires sont `psych` et `GPArotation`. Le fichier de syntaxe R Chapitre 11 – Analyse factorielle exploratoire – Input .R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")
```

```
# Sélectionner le(s) librairie(s) nécessaires pour l'analyse
library(psych)
library(GPArotation)
```

```
# Importer la base de données
data <- read.csv("efa_data.csv", header = TRUE, sep = ",")
```

```
# Inspecter la base de données
str(data)
```

```
# Déterminer les valeurs propres
eigen(cor(data))
```

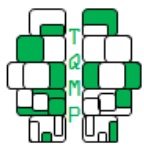
```
# Définir le modèle
# Modèle à 2 facteurs
model.2 <- fa(data, nfactors = 2, rotate = "geominQ", fm = "ml")
```

```
# Obtenir le sommaire des estimations du modèle
model.2
```

```
# Obtenir le CFI
((model.2$null.chisq - model.2$null.dof) - (model.2$STATISTIC - model.2$dof))
  / (model.2$null.chisq - model.2$null.dof)
```

Output 3.2. Sortie de l'analyse factorielle exploratoire du Listing 3.2. Les nombres en vert représentent la structure factorielle après rotations puis les nombres en jaune sont les indices d'ajustement.

```
> # Obtenir le sommaire des estimations du modèle
> model.2
```



Factor Analysis using method = ml

Call: fa(r = data, nfactors = 2, rotate = "geominQ", fm = "ml")

Standardized loadings (pattern matrix) based upon correlation matrix

	ML1	ML2	h2	u2	com
I1	0.03	0.97	0.94	0.058	1.0
I2	0.01	0.93	0.87	0.130	1.0
I3	0.01	0.78	0.61	0.389	1.0
I4	-0.06	0.65	0.44	0.562	1.0
I5	-0.09	0.63	0.42	0.584	1.0
I6	0.98	-0.02	0.97	0.032	1.0
I7	0.92	0.00	0.84	0.161	1.0
I8	0.79	0.05	0.62	0.375	1.0
I9	0.75	-0.04	0.57	0.428	1.0
I10	0.53	0.10	0.28	0.719	1.1

	ML1	ML2
SS loadings	3.29	3.27
Proportion Var	0.33	0.33
Cumulative Var	0.33	0.66
Proportion Explained	0.50	0.50
Cumulative Proportion	0.50	1.00

With factor correlations of

	ML1	ML2
ML1	1.00	-0.07
ML2	-0.07	1.00

Mean item complexity = 1

Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are 45 and the objective function was 7.57 with Chi Square of 1474.14

The degrees of freedom for the model are 26 and the objective function was 0.17

The root mean square of the residuals (RMSR) is 0.02

The df corrected root mean square of the residuals is 0.03

The harmonic number of observations is 200 with the empirical chi square 11.06 with prob < 1

The total number of observations was 200 with Likelihood Chi Square = 32.79 with prob < 0.17

Tucker Lewis Index of factoring reliability = 0.992

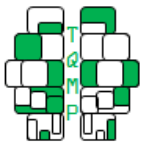
RMSEA index = 0.036 and the 90 % confidence intervals are 0 0.07

BIC = -104.97

Fit based upon off diagonal values = 1

Measures of factor score adequacy

ML1	ML2
-----	-----



```
Correlation of (regression) scores with factors    0.99 0.98
Multiple R square of scores with factors          0.97 0.96
Minimum correlation of possible factor scores      0.95 0.93
```

```
> # Obtenir le CFI
> ((model.2$null.chisq - model.2$null.dof) - (model.2$STATISTIC - model.2$dof))
  / (model.2$null.chisq - model.2$null.dof

[1] 0.9952506
```

Listing 3.3. Syntaxe pour réaliser une analyse factorielle exploratoire avec un modèle à trois facteurs (modèle 3). Le fichier de données utilisé est `efa_data.csv`, les variables utilisées sont I1 à I10, puis les librairies nécessaires sont `psych` et `GPArotation`. Le fichier de syntaxe R Chapitre 11 – Analyse factorielle exploratoire – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaires pour l'analyse
library(psych)
library(GPArotation)

# Importer la base de données
data <- read.csv("efa_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Déterminer les valeurs propres
eigen(cor(data))

# Définir le modèle
# Modèle à 3 facteurs
model.3 <- fa(data, nfactors = 3, rotate = "geominQ", fm = "ml")

# Obtenir le sommaire des estimations du modèle
model.3

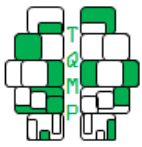
# Obtenir le CFI
((model.3$null.chisq - model.3$null.dof) - (model.3$STATISTIC - model.3$dof))
  / (model.3$null.chisq - model.3$null.dof)
```

Output 3.3. Sortie de l'analyse factorielle exploratoire du Listing 3.3. Les nombres en **vert** représentent la structure factorielle après rotations puis les nombres en **jaune** sont les indices d'ajustement.

```
> # Obtenir le sommaire des estimations du modèle
> model.3

Factor Analysis using method = ml
Call: fa(r = data, nfactors = 3, rotate = "geominQ", fm = "ml")

Standardized loadings (pattern matrix) based upon correlation matrix
  ML1  ML2  ML3  h2  u2 com
```



I1	0.05	0.97	-0.01	0.94	0.063	1.0
I2	0.04	0.94	-0.03	0.87	0.126	1.0
I3	0.02	0.77	0.18	0.65	0.351	1.1
I4	-0.07	0.63	0.29	0.53	0.473	1.4
I5	-0.06	0.67	-0.31	0.52	0.482	1.4
I6	0.98	-0.02	-0.02	0.97	0.034	1.0
I7	0.91	-0.01	0.10	0.85	0.146	1.0
I8	0.79	0.04	0.04	0.63	0.372	1.0
I9	0.76	-0.03	-0.11	0.58	0.415	1.0
I10	0.53	0.11	-0.03	0.28	0.717	1.1

	ML1	ML2	ML3
SS loadings	3.29	3.28	0.25
Proportion Var	0.33	0.33	0.02
Cumulative Var	0.33	0.66	0.68
Proportion Explained	0.48	0.48	0.04
Cumulative Proportion	0.48	0.96	1.00

With factor correlations of

	ML1	ML2	ML3
ML1	1.00	-0.10	0.06
ML2	-0.10	1.00	0.09
ML3	0.06	0.09	1.00

Mean item complexity = 1.1

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are 45 and the objective function was 7.57 with Chi Square of 1474.14

The degrees of freedom for the model are 18 and the objective function was 0.09

The root mean square of the residuals (RMSR) is 0.01

The df corrected root mean square of the residuals is 0.02

The harmonic number of observations is 200 with the empirical chi square 3.12 with prob < 1

The total number of observations was 200 with Likelihood Chi Square = 16.47 with prob < 0.56

Tucker Lewis Index of factoring reliability = 1.003

RMSEA index = 0 and the 90 % confidence intervals are 0 0.058

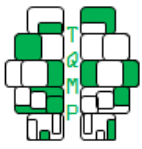
BIC = -78.9

Fit based upon off diagonal values = 1

Measures of factor score adequacy

	ML1	ML2	ML3
Correlation of (regression) scores with factors	0.99	0.98	0.61
Multiple R square of scores with factors	0.97	0.96	0.38
Minimum correlation of possible factor scores	0.95	0.92	-0.24

> # Obtenir le CFI



```
> ((model.3$null.chisq - model.3$null.dof) - (model.3$STATISTIC - model.3$dof))
  / (model.3$null.chisq - model.3$null.dof)
```

```
[1] 1.001069
```

Annexe 4 : L'analyse factorielle confirmatoire (*Confirmatory factor analysis*)

Listing 4.1. Syntaxe pour réaliser une analyse factorielle confirmatoire avec les items de référence par défaut (modèle 1). Dans cet exemple, les items de références sont I1 et I6. Le fichier de données utilisé est `cfa_data.csv`, les variables utilisées sont I1 à I10, puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 12 – Analyse factorielle confirmatoire – Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("cfa_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Définir le modèle
# Modèle 1 (items de référence par défaut)
model.1 <- 'Factor 1 =~ I1 + I2 + I3 + I4 + I5
  Factor 2 =~ I6 + I7 + I8 + I9 + I10'

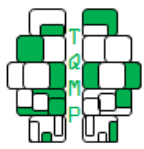
# Spécifier les paramètres du modèle
fit.1 <- cfa(model.1, data = data)

# Obtenir le sommaire des estimations du modèle
summary(fit.1, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 4.1. Sortie de l'analyse factorielle confirmatoire du Listing 4.1. Les nombres en **jaune** représentent les indices d'ajustement, les nombres en **gris** sont les items de référence par défaut, les nombres en **vert** sont les coefficients et leur valeur *p*, les nombres en **bleu** sont les coefficients standardisés, les nombres en **magenta** sont la variance expliquée par chacun des items.

```
> # Inspecter la base de données
> str(data)

'data.frame': 200 obs. of 10 variables:
 $ I1 : int  9 10 6 9 12 3 5 3 14 -7 ...
 $ I2 : int  8 10 6 8 11 4 4 1 15 -9 ...
 $ I3 : int  5 6 9 6 6 3 3 4 11 -9 ...
 $ I4 : int  8 -1 3 9 9 6 6 3 12 -1 ...
 $ I5 : int  6 10 2 5 8 5 4 10 15 -8 ...
 $ I6 : int  6 2 1 14 -3 -1 6 2 5 5 ...
 $ I7 : int  3 2 2 14 -2 -1 5 -1 5 8 ...
 $ I8 : int  9 4 4 14 -3 1 5 1 4 5 ...
 $ I9 : int  7 1 2 13 -3 -1 5 5 4 10 ...
 $ I10: int -1 6 2 15 3 -2 2 4 4 1 ...
```



```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.1, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

lavaan 0.6-7 ended normally after 61 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	21
Number of observations	200

Model Test User Model:

Test statistic	46.084
Degrees of freedom	34
P-value (Chi-square)	0.081

Model Test Baseline Model:

Test statistic	1513.237
Degrees of freedom	45
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.992
Tucker-Lewis Index (TLI)	0.989

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-5097.533
Loglikelihood unrestricted model (H1)	-5074.491

Akaike (AIC)	10237.066
Bayesian (BIC)	10306.330
Sample-size adjusted Bayesian (BIC)	10239.800

Root Mean Square Error of Approximation:

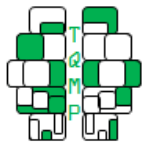
RMSEA	0.042
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.071
P-value RMSEA <= 0.05	0.641

Standardized Root Mean Square Residual:

SRMR	0.044
------	-------

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
Factor1 =~						
I1	1.000				4.652	0.968
I2	0.894	0.033	26.742	0.000	4.157	0.935
I3	0.720	0.045	16.171	0.000	3.349	0.783
I4	0.641	0.055	11.675	0.000	2.983	0.658
I5	0.676	0.061	11.064	0.000	3.146	0.636
Factor2 =~						



I6	1.000				4.622	0.984
I7	0.821	0.031	26.395	0.000	3.794	0.916
I8	0.721	0.043	16.825	0.000	3.331	0.788
I9	0.701	0.046	15.358	0.000	3.242	0.756
I10	0.491	0.059	8.347	0.000	2.268	0.518

Covariances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
Factor1 ~~						
Factor2	-1.525	1.575	-0.969	0.333	-0.071	-0.071

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.I1	1.461	0.448	3.260	0.001	1.461	0.063
.I2	2.485	0.424	5.865	0.000	2.485	0.126
.I3	7.092	0.762	9.307	0.000	7.092	0.387
.I4	11.646	1.203	9.679	0.000	11.646	0.567
.I5	14.548	1.497	9.716	0.000	14.548	0.595
.I6	0.722	0.396	1.826	0.068	0.722	0.033
.I7	2.760	0.385	7.171	0.000	2.760	0.161
.I8	6.767	0.719	9.409	0.000	6.767	0.379
.I9	7.866	0.824	9.543	0.000	7.866	0.428
.I10	13.994	1.414	9.896	0.000	13.994	0.731
Factor1	21.639	2.344	9.232	0.000	1.000	1.000
Factor2	21.366	2.242	9.532	0.000	1.000	1.000

R-Square:

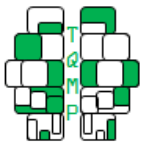
	Estimate
I1	0.937
I2	0.874
I3	0.613
I4	0.433
I5	0.405
I6	0.967
I7	0.839
I8	0.621
I9	0.572
I10	0.269

Listing 4.2. Syntaxe pour réaliser une analyse factorielle confirmatoire avec les items de référence spécifiés (modèle 2). Dans cet exemple, les items de références spécifiés sont I2 et I7. Le fichier de données utilisé est `cfa_data.csv`, les variables utilisées sont I1 à I10, puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 12 – Analyse factorielle confirmatoire – Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
```



```
data <- read.csv("cfa_data.csv", header = TRUE, sep = ",")
```

```
# Inspecter la base de données
str(data)
```

```
# Définir le modèle
# Modèle 2 (items de référence spécifiés*)
model.2 <- 'Factor 1 =~ NA*I1 + 1*I2 + I3 + I4 + I5
           Factor 2 =~ NA*I6 + 1*I7 + I8 + I9 + I10'
```

```
# Spécifier les paramètres du modèle
fit.2 <- cfa(model.2, data = data)
```

```
# Obtenir le sommaire des estimations du modèle
summary(fit.2, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 4.2. Sortie de l'analyse factorielle confirmatoire du Listing 4.2. Les nombres en **jaune** représentent les indices d'ajustement, les nombres en **gris** sont les items de référence spécifiés, les nombres en **vert** sont les coefficients et leur valeur p , les nombres en **bleu** sont les coefficients standardisés, les nombres en **magenta** sont la variance expliquée par chacun des items.

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.2, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

```
lavaan 0.6-7 ended normally after 57 iterations
```

Estimator	ML
Optimization method	NLMINB
Number of free parameters	21
Number of observations	200

```
Model Test User Model:
```

Test statistic	46.084
Degrees of freedom	34
P-value (Chi-square)	0.081

```
Model Test Baseline Model:
```

Test statistic	1513.237
Degrees of freedom	45
P-value	0.000

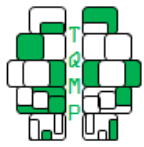
```
User Model versus Baseline Model:
```

Comparative Fit Index (CFI)	0.992
Tucker-Lewis Index (TLI)	0.989

```
Loglikelihood and Information Criteria:
```

Loglikelihood user model (H0)	-5097.533
Loglikelihood unrestricted model (H1)	-5074.491

Akaike (AIC)	10237.066
Bayesian (BIC)	10306.330
Sample-size adjusted Bayesian (BIC)	10239.800



Root Mean Square Error of Approximation:

RMSEA	0.042
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.071
P-value RMSEA ≤ 0.05	0.641

Standardized Root Mean Square Residual:

SRMR	0.044
------	-------

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
Factor1 =~						
I1	1.119	0.042	26.742	0.000	4.652	0.968
I2	1.000				4.157	0.935
I3	0.806	0.052	15.614	0.000	3.349	0.783
I4	0.718	0.063	11.454	0.000	2.983	0.658
I5	0.757	0.070	10.875	0.000	3.146	0.636
Factor2 =~						
I6	1.218	0.046	26.395	0.000	4.622	0.984
I7	1.000				3.794	0.916
I8	0.878	0.056	15.554	0.000	3.331	0.788
I9	0.854	0.059	14.368	0.000	3.242	0.756
I10	0.598	0.073	8.174	0.000	2.268	0.518

Covariances:

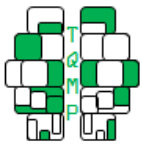
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
Factor1 ~~						
Factor2	-1.119	1.156	-0.968	0.333	-0.071	-0.071

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.I1	1.461	0.448	3.260	0.001	1.461	0.063
.I2	2.485	0.424	5.865	0.000	2.485	0.126
.I3	7.092	0.762	9.307	0.000	7.092	0.387
.I4	11.646	1.203	9.679	0.000	11.646	0.567
.I5	14.548	1.497	9.716	0.000	14.548	0.595
.I6	0.722	0.396	1.826	0.068	0.722	0.033
.I7	2.760	0.385	7.171	0.000	2.760	0.161
.I8	6.767	0.719	9.409	0.000	6.767	0.379
.I9	7.866	0.824	9.543	0.000	7.866	0.428
.I10	13.994	1.414	9.896	0.000	13.994	0.731
Factor1	17.279	1.991	8.680	0.000	1.000	1.000
Factor2	14.394	1.714	8.397	0.000	1.000	1.000

R-Square:

	Estimate
I1	0.937
I2	0.874
I3	0.613
I4	0.433



I5	0.405
I6	0.967
I7	0.839
I8	0.621
I9	0.572
I10	0.269

Annexe 5 : L'analyse de médiation (*Mediation analysis*)

Listing 5.1. Syntaxe pour réaliser une analyse de médiation pour un modèle sans ré-échantillonnage (modèle 1). Le fichier de données utilisé est `mediation_data.csv`, les variables utilisées sont VI, VM et VD, puis les librairies nécessaires sont `lavaan`, `boot`, et `mice`. Le fichier de syntaxe R Chapitre 13 – L'analyse de médiation – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)
library(boot)
library(mice)

# Importer la base de données
data <- read.csv("mediation_data.csv", header = TRUE, sep = ",")

# Remplacer les valeurs manquantes codées 999 (Mplus) par NA
data[data == 999] <- NA

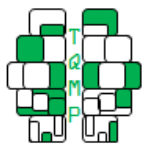
# Inspecter la base de données
str(data)
md.pattern(data)

# Définir le modèle
# Modèle 1 (modèle sans rééchantillonnage)
model.1 <- 'VD ~ c.prime*VI
            VM ~ a*VI
            VD ~ b*VM
            ab := a*b
            c  := c.prime + (a*b)

# Spécifier les paramètres du modèle
fit.1 <- sem(model.1, data = data, missing = "fiml")

# Obtenir le sommaire des estimations du modèle
summary(fit.1, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 5.1. Sortie de l'analyse de médiation du Listing 5.1. Les nombres en gris sont le total des observations manquantes, les nombres en magenta sont les observations avec des données manquantes pour une variable ou plus, les nombres en vert sont les coefficients de régression et leur valeur p , les nombres en rouge sont les coefficients de régression standardisés, les nombres en bleu sont la variance expliquée par la variable dépendante et médiatrice puis les nombres en jaune sont les paramètres de l'effet indirect (ab) et total (c).



```
> # Inspecter la base de données
> str(`data
```

```
'data.frame': 208 obs. of 3 variables:
 $ VM: num NA NA NA NA NA NA NA NA NA NA ...
 $ VI: int 1 0 1 0 0 1 0 1 1 1 ...
 $ VD: num -0.481 -0.628 -0.628 -0.481 -0.481 ...
```

```
> md.pattern(data)
```

	VI	VD	VM
165	1	1	0
41	1	1	0
1	1	0	1
1	1	0	0
	0	2	42

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.1, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

lavaan 0.6-7 ended normally after 22 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	7
Number of observations	208
Number of missing patterns	4

Model Test User Model:

Test statistic	0.000
Degrees of freedom	0

Model Test Baseline Model:

Test statistic	21.783
Degrees of freedom	3
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	1.000
Tucker-Lewis Index (TLI)	1.000

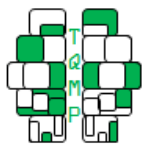
Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-410.061
Loglikelihood unrestricted model (H1)	-410.061

Akaike (AIC)	834.122
Bayesian (BIC)	857.485
Sample-size adjusted Bayesian (BIC)	835.305

Root Mean Square Error of Approximation:

RMSEA	0.000
-------	-------



```

90 Percent confidence -intervl - lower      0.000
90 Percent confidence -intervl - upper      0.000
P-value RMSEA <= 0.05                      NA

```

```

Standardized Root Mean Square Residual:
SRMR                                     0.000

```

Regressions:

		Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
VD ~							
VI	(c.pr)	0.392	0.149	2.622	0.009	0.392	0.180
VM ~							
VI	(a)	0.220	0.089	2.475	0.013	0.220	0.192
VD ~							
VM	(b)	0.360	0.149	2.410	0.016	0.360	0.190

Intercepts:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.VD	-0.660	0.231	-2.862	0.004	-0.660	-0.661
.VM	1.439	0.047	30.341	0.000	1.439	2.727

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.VD	0.913	0.091	10.008	0.000	0.913	0.918
.VM	0.268	0.030	9.074	0.000	0.268	0.963

R-Square:

	Estimate
VD	0.082
VM	0.037

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
ab	0.079	0.045	1.753	0.080	0.079	0.036
c	0.471	0.148	3.183	0.001	0.471	0.217

Listing 5.2. Syntaxe pour réaliser une analyse de médiation pour un modèle avec ré-échantillonnage (modèle 2). Le fichier de données utilisé est `mediation_data.csv`, les variables utilisées sont VI, VM et VD, puis les librairies nécessaires sont `lavaan`, `boot`, et `mice`. Le fichier de syntaxe R Chapitre 13 – L'analyse de médiation – `Input.R` se trouve dans le matériel accompagnateur.

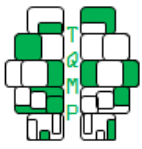
```

# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)
library(boot)
library(mice)

# Importer la base de données
data <- read.csv("mediation_data.csv", header = TRUE, sep = ",")

```



```
# Remplacer les valeurs manquantes codées 999 (MPlus) par NA (R)
data[data == 999] <- NA

# Inspecter la base de données
str(data)
md.pattern(data)

# Définir le modèle
# Modèle 2 (modèle avec rééchantillonnage)
model.2 <- 'VD ~ c.prime*VI
            VM ~ a*VI
            VD ~ b*VM
            ab := a*b
            c := c.prime + (a*b)'

# Spécifier les paramètres du modèle
fit.2 <- sem(model.2, data = data, missing = "fiml", se = "bootstrap", bootstrap =
5000)

# Obtenir le sommaire des estimations du modèle
summary(fit.2, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)

parameterEstimates(fit, boot.ci.type = "norm", level = 0.95,
standardized = TRUE)
```

Output 5.2. Sortie de l'analyse de médiation du Listing 5.2. Les nombres en **vert** sont les coefficients de régression et leur valeur p , les nombres en **rouge** sont les coefficients de régression standardisés, les nombres en **bleu** sont la variance expliquée par la variable dépendante et médiatrice puis les nombres en **jaune** sont les paramètres de l'effet indirect (ab) et total (c) puis les nombres en **gris** sont les intervalles de confiance bootstrap.

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.2, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

lavaan 0.6-7 ended normally after 22 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	7
Number of observations	208
Number of missing patterns	4

Model Test User Model:

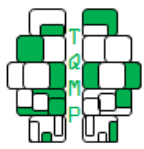
Test statistic	0.000
Degrees of freedom	0

Model Test Baseline Model:

Test statistic	21.783
Degrees of freedom	3
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	1.000
Tucker-Lewis Index (TLI)	1.000



Loglikelihood and Information Criteria:

Loglikelihood user <code>model</code> (H0)	-410.061
Loglikelihood unrestricted <code>model</code> (H1)	-410.061
Akaike (AIC)	834.122
Bayesian (BIC)	857.485
Sample-size adjusted Bayesian (BIC)	835.305

Root Mean Square Error of Approximation:

RMSEA	0.000
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.000
P-value RMSEA $\leq$ 0.05	NA

Standardized Root Mean Square Residual:

SRMR	0.000
------	-------

Regressions:

		Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper	Std.lv	Std.all
VD ~									
VI (c.pr)		0.392	0.170	2.308	0.021	0.056	0.723	0.392	0.180
VM ~									
VI (a)		0.220	0.086	2.546	0.011	0.058	0.389	0.220	0.192
VD ~									
VM (b)		0.360	0.140	2.579	0.010	0.100	0.637	0.360	0.190

Intercepts:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper	Std.lv	Std.all
.VD	-0.660	0.179	-3.679	0.000	-1.018	-0.319	-0.660	-0.661
.VM	1.439	0.049	29.111	0.000	1.341	1.535	1.439	2.727

Variances:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper	Std.lv	Std.all
.VD	0.913	0.128	7.119	0.000	0.657	1.161	0.913	0.918
.VM	0.268	0.027	9.861	0.000	0.213	0.320	0.268	0.963

R-Square:

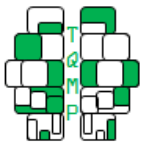
	Estimate
VD	0.082
VM	0.037

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)	ci.lower	ci.upper	Std.lv	Std.all
ab	0.079	0.039	2.038	0.042	0.014	0.161	0.079	0.036
c	0.471	0.164	2.878	0.004	0.150	0.795	0.471	0.217

```
> parameterEstimates(fit.2, boot.ci.type = "norm", level = 0.95,
  standardized = TRUE)
```

Bootstrap confidence intervals:



lhs	op	rhs	label	est	se	z	pvalue	ci.low	ci.up	std.lv	std.all	std.nox
VD	~	VI	c.prime	0.392	0.170	2.308	0.021	0.054	0.719	0.392	0.180	0.393
VM	~	VI	a	0.220	0.086	2.546	0.011	0.047	0.386	0.220	0.192	0.417
VD	~	VM	b	0.360	0.140	2.579	0.010	0.085	0.632	0.360	0.190	0.190
VD	~~	VD		0.913	0.128	7.119	0.000	0.677	1.180	0.913	0.918	0.918
VM	~~	VM		0.268	0.027	9.861	0.000	0.218	0.325	0.268	0.963	0.963
VI	~~	VI		0.211	0.000	NA	NA	0.211	0.211	0.211	1.000	0.211
VD	~1			-0.660	0.179	-3.679	0.000	-1.008	-0.305	-0.660	-0.661	-0.661
VM	~1			1.439	0.049	29.111	0.000	1.343	1.537	1.439	2.727	2.727
VI	~1			0.303	0.000	NA	NA	0.303	0.303	0.303	0.659	0.303
ab	:=	a*b	ab	0.079	0.039	2.038	0.042	0.002	0.154	0.079	0.036	0.079
c	:=	c.prime+ab	c	0.471	0.164	2.878	0.004	0.144	0.785	0.471	0.217	0.472

Annexe 6 : L'analyse de modération (*Moderation analysis*)

Listing 6.1. Syntaxe pour réaliser une analyse de modération avec la librairie `gvlma` (modèle 1). Le fichier de données utilisé est `moderation_data.csv`, les variables utilisées sont VI, VM, VD et VIVM puis les librairies nécessaires sont `mice`, `gvlma`, et `lavaan`. Le fichier de syntaxe R Chapitre 14 – L'analyse de modération – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd( "chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library (gvlma)
library (lavaan)
library(mice)

# Importer la base de données
data <- read.csv("moderation_data.csv", header = TRUE, sep = ",")

# Remplacer les valeurs manquantes codées 999 (MPlus) par NA (R)
data[data == 999] <- NA

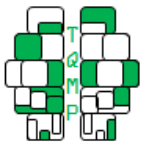
# Inspecter la base de données
str(data)
md.pattern(data)

# Modifier la base de données
# Créer la variable VIVM et exclure les données manquantes
data$VIVM <- data$VI * data$VM

# Exclure les données
data <- data[!(is.na(data$VI) | is.na(data$VD) | is.na(data$VIVM)),]

# Analyses descriptives
# Mesures de tendance centrale
summary (data)

# Covariances
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))
```



```
# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
model <- 'VD ~ VI + VM + VIVM'

# Spécifier les paramètres du modèle
fit.1 <- lm(model, data = data)

# Obtenir le sommaire des estimations du modèle
summary(fit.1, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 6.1. Sortie de l'analyse de modération du Listing 6.1. Les nombres en gris représentent le total des observations manquantes pour chaque variable, les nombres en bleu représentent les observations avec des données manquantes pour une variable ou plus, les nombres en jaune sont les moyennes pour chacune des variables, les nombres en magenta sont les variances de chaque variable, les nombres en vert sont les coefficients de régression et leur valeur p puis les nombres en rouge représentent la variance expliquée par la variable dépendante.

```
> # Inspecter la base de données
> str(data)

'data.frame': 208 obs. of 3 variables:
 $ VI: num NA NA NA NA NA NA NA NA NA NA ...
 $ VD: num NA NA NA NA NA NA NA NA NA NA ...
 $ VM: int 0 0 0 0 1 0 1 1 0 0 ...
```

```
> md.pattern(data)
```

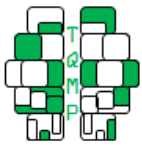
	VM	VI	VD
166	1	1	0
42	1	0	2
	0	42	42

```
> # Analyses descriptives
> # Mesures de tendance centrale
> summary (data)
```

VI		VD		VM		VIVM	
Min.	:0.08991	Min.	:0.0000	Min.	:0.0000	Min.	:0.0000
1st Qu.	:1.11385	1st Qu.	:0.0000	1st Qu.	:0.0000	1st Qu.	:0.0000
Median	:1.51675	Median	:0.9348	Median	:0.0000	Median	:0.0000
Mean	:1.49566	Mean	:0.8661	Mean	:0.4759	Mean	:0.7314
3rd Qu.	:1.85615	3rd Qu.	:1.5887	3rd Qu.	:1.0000	3rd Qu.	:1.4805
Max.	:2.85666	Max.	:2.8567	Max.	:1.0000	Max.	:2.8567

```
> # Covariances
> cov(data, y = NULL, method = c("pearson"))
```

	VI	VD	VM	VIVM
VI	0.2793200	0.1915164	0.0197040	0.1579549



```

VD    0.1915164 0.7402413 0.3211141 0.6211671
VM    0.0197040 0.3211141 0.2509310 0.3856351
VIVM  0.1579549 0.6211671 0.3856351 0.7203241

```

```

> # Correlations
> cor(data, y = NULL, method = c("pearson"))

```

```

          VI          VD          VM          VIVM
VI    1.00000000 0.4211806 0.07442626 0.3521422
VD    0.42118061 1.0000000 0.74506740 0.8506636
VM    0.07442626 0.7450674 1.00000000 0.9070587
VIVM  0.35214218 0.8506636 0.90705875 1.0000000

```

```

> # Obtenir le sommaire des estimations du modèle
> summary(fit.1, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)

```

```

Call:
lm(formula = model, data = data)

```

```

Residuals:
    Min       1Q   Median       3Q      Max
-0.5642 -0.2553  0.0000  0.0000  2.1616

```

```

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.11816    0.13788  -0.857  0.39269
VI           0.25735    0.08879   2.898  0.00427 **
VM           0.11816    0.20826   0.567  0.57123
VIVM         0.74265    0.13097   5.671  6.38e-08 ***
---

```

```

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

```

```

Residual standard error: 0.4419 on 162 degrees of freedom

```

```

Multiple R-squared:  0.741,    Adjusted R-squared:  0.7362

```

```

F-statistic: 154.5 on 3 and 162 DF,  p-value: < 2.2e-16

```

Listing 6.2. Syntaxe pour réaliser une analyse de modération avec la librairie lavaan (modèle 2). Le fichier de données utilisé est `moderation_data.csv`, les variables utilisées sont VI, VM, VD et VIVM puis les librairies nécessaires sont `mice`, `gvlma`, et `lavaan`. Le fichier de syntaxe R Chapitre 14 – L’analyse de modération – `Input.R` se trouve dans le matériel accompagnateur.

```

# Définir le répertoire de travail
setwd( "chemin du répertoire dans lequel la base de données a été sauvegardée")

```

```

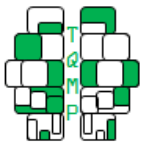
# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library (gvlma)
library (lavaan)
library(mice)

```

```

# Importer la base de données
data <- read.csv("moderation_data.csv", header = TRUE, sep = ",")

```



```

# Remplacer les valeurs manquantes codées 999 (MPlus) par NA (R)
data[data == 999] <- NA

# Inspecter la base de données
str(data)
md.pattern(data)

# Modifier la base de données
# Créer la variable VIVM et exclure les données manquantes
data$VIVM <- data$VI * data$VM

# Exclure les données
data <- data[!(is.na(data$VI) | is.na(data$VD) | is.na(data$VIVM)),]

# Analyses descriptives
# Mesures de tendance centrale
summary (data)

# Covariances
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
model <- 'VD ~ VI + VM + VIVM'

# Spécifier les paramètres du modèle
fit.2 <- sem(model, data = data)

# Obtenir le sommaire des estimations du modèle
summary(fit.2, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)

```

Output 6.2. Sortie de l'analyse de modération du Listing 6.2. Les nombres en **vert** sont les coefficients de régression et leur valeur *p*, les nombres en **gris** sont les coefficients de régression standardisés puis le nombre en **bleu** représente la variance expliquée par la variable dépendante.

```

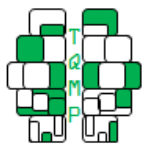
> # Obtenir le sommaire des estimations du modèle
> summary(fit.2, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)

lavaan 0.6-7 ended normally after 26 iterations
  Estimator                               ML
  Optimization method                     NLMINB
  Number of free parameters                4
  Number of observations                   166

Model Test User Model:
  Test statistic                           0.000
  Degrees of freedom                       0

Model Test Baseline Model:
  Test statistic                           224.273
  Degrees of freedom                       3

```

P-value 0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI) 1.000

Tucker-Lewis Index (TLI) 1.000

Loglikelihood and Information Criteria:

Loglikelihood user model (H0) -97.941

Loglikelihood unrestricted model (H1) -97.941

Akaike (AIC) 203.882

Bayesian (BIC) 216.330

Sample-size adjusted Bayesian (BIC) 203.666

Root Mean Square Error of Approximation:

RMSEA 0.000

90 Percent confidence interval - lower 0.000

90 Percent confidence interval - upper 0.000

P-value RMSEA <= 0.05 NA

Standardized Root Mean Square Residual:

SRMR 0.000

Regressions:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
VD ~						
VI	0.257	0.088	2.934	0.003	0.257	0.158
VM	0.118	0.206	0.574	0.566	0.118	0.069
VIVM	0.743	0.129	5.740	0.000	0.743	0.733

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.VD	0.191	0.021	9.110	0.000	0.191	0.259

R-Square:

	Estimate
VD	0.741

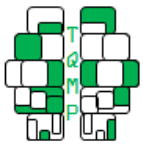
Annexe 7 : L'analyse de médiation modérée (*Moderated mediation analysis*)

Listing 7.1. Syntaxe pour réaliser une analyse de médiation modérée avec un modérateur dichotomique (modèle 1). Le fichier de données utilisé est `med_mod_data_1.csv`, les variables utilisées sont X, M, V, Y et MV puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 15 - L'analyse de médiation modérée - Input .R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("mod_med_data_1.csv", header = TRUE, sep = ",")
```



```
# Inspecter la base de données
str(data)

# Modifier la base de données
# Créer la variable MV
data$MV <- data$M * data$V

# Analyses descriptives
# Mesures de tendance centrale
summary(data)

# Covariance
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
# Modèle 1 (modèle avec modérateur dichotomique)
model.1 = 'Y ~ cprime * X + b * M + V + v * MV
          M ~ a * X
          ab   := a * b
          abv  := a * (b + v)
          ctot := ab + cprime
          cvtot:= abv + cprime
          d    := ab - abv'

# Spécifier les paramètres du modèle
fit <- sem(model.1, data = data, missing = "listwise", estimator = "ML")

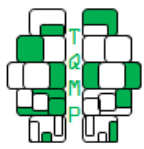
# Obtenir le sommaire des estimations du modèle
summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 7.1. Sortie de l'analyse de médiation modérée du Listing 7.1. Les nombres en magenta sont les moyennes de chaque variable, les nombres en vert sont les coefficients de régression et leur valeur p, les nombres en gris sont les coefficients de régression standardisés, les nombres en bleu représentent la variance expliquée par la variable dépendante et médiatrice puis les nombres en jaune représentent l'estimée des paramètres créés et inclus dans le modèle.

```
> # Inspecter la base de données
> str(data)

'data.frame': 200 obs. of 4 variables:
 $ X: num 0.07 -0.87 -0.36 0.72 0.26 0.39 -0.7 -1.08 0.24 0.21 ...
 $ M: num -0.43 -1.42 -0.6 -0.73 1.04 -2.11 -0.25 0.03 1.43 0.63 ...
 $ V: int 1 1 0 1 1 1 0 1 0 0 ...
 $ Y: num -1.02 -1.95 1.94 -0.72 0.59 -2.01 0.21 -0.65 0.5 2.69 ...

> # Analyses descriptives
> # Mesures de tendance centrale
> summary(data)
```



X	M	V	Y
Min. : -2.70000	Min. : -2.33000	Min. : 0.00	Min. : -3.26000
1st Qu.: -0.58500	1st Qu.: -0.70000	1st Qu.: 0.00	1st Qu.: -0.80750
Median : 0.11000	Median : 0.01500	Median : 0.00	Median : 0.14000
Mean : 0.04195	Mean : 0.03185	Mean : 0.47	Mean : 0.03825
3rd Qu.: 0.54500	3rd Qu.: 0.75250	3rd Qu.: 1.00	3rd Qu.: 0.63000
Max. : 2.89000	Max. : 2.48000	Max. : 1.00	Max. : 3.27000

MV
Min. : -2.3300
1st Qu.: 0.0000
Median : 0.0000
Mean : -0.0435
3rd Qu.: 0.0000
Max. : 2.0800

```
> # Covariances
> cov(data, y = NULL, use = "complete.obs", method = c("pearson"))
```

	X	M	V	Y	MV
X	0.87903487	0.30942049	-0.04549397	0.27389288	0.17174204
M	0.30942049	0.95262319	-0.05876332	0.59552637	0.48910902
V	-0.04549397	-0.05876332	0.25035176	-0.07580653	-0.02317085
Y	0.27389288	0.59552637	-0.07580653	1.18178335	0.41159033
MV	0.17174204	0.48910902	-0.02317085	0.41159033	0.48581482

```
> # Corrélations
> cov(data, y = NULL, use = "complete.obs", method = c("pearson"))
```

	X	M	V	Y	MV
X	1.0000000	0.3381314	-0.09697850	0.2687254	0.26280775
M	0.3381314	1.0000000	-0.12032903	0.5612696	0.71896826
V	-0.0969785	-0.1203290	1.00000000	-0.1393678	-0.06644026
Y	0.2687254	0.5612696	-0.13936778	1.0000000	0.54320135
MV	0.2628078	0.7189683	-0.06644026	0.5432014	1.00000000

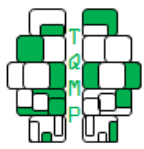
```
> # Obtenir le sommaire des estimations du modèle
> summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

lavaan 0.6-7 ended normally after 14 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	7
Number of observations	200

Model Test User Model:	
Test statistic	132.963
Degrees of freedom	2
P-value (Chi-square)	0.000

Model Test Baseline Model:



Test statistic	248.666
Degrees of freedom	7
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.458
Tucker-Lewis Index (TLI)	-0.897

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-520.570
Loglikelihood unrestricted model (H1)	-454.089

Akaike (AIC)	1055.140
Bayesian (BIC)	1078.229
Sample-size adjusted Bayesian (BIC)	1056.052

Root Mean Square Error of Approximation:

RMSEA	0.572
90 Percent confidence interval - lower	0.492
90 Percent confidence interval - upper	0.657
P-value RMSEA ≤ 0.05	0.000

Standardized Root Mean Square Residual:

SRMR	0.182
------	-------

Regressions:

		Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
Y ~							
X	(cprm)	0.090	0.072	1.260	0.208	0.090	0.083
M	(b)	0.354	0.067	5.322	0.000	0.354	0.339
V		-0.161	0.123	-1.313	0.189	-0.161	-0.079
MV	(v)	0.451	0.091	4.952	0.000	0.451	0.308
M ~							
X	(a)	0.352	0.069	5.081	0.000	0.352	0.338

Variances:

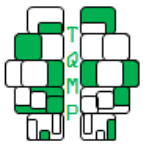
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.Y	0.744	0.074	10.000	0.000	0.744	0.719
.M	0.839	0.084	10.000	0.000	0.839	0.886

R-Square:

	Estimate
Y	0.281
M	0.114

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
ab	0.125	0.034	3.675	0.000	0.125	0.115
abv	0.283	0.068	4.140	0.000	0.283	0.219
ctot	0.215	0.072	2.982	0.003	0.215	0.198



cvtot	0.374	0.087	4.283	0.000	0.374	0.302
d	-0.159	0.045	-3.546	0.000	-0.159	-0.104

Listing 7.2. Syntaxe pour réaliser une analyse de médiation modérée avec un modérateur continu (modèle 2). Le fichier de données utilisé est `med_mod_data_2.csv`, les variables utilisées sont X, M, V, Y et MV puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 15 – L’analyse de médiation modérée – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("mod_med_data_2.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Modifier la base de données
# Créer la variable MV
data$MV <- data$M * data$V

# Analyses descriptives
# Mesures de tendance centrale
summary(data)

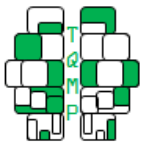
# Covariances
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
# Modèle 2 (modèle avec modérateur continu)
model.2 <- 'Y ~ cprime * X + b * M + V + v * MV
           M ~ a * X
           low    := -1
           med     := 0
           hig     := +1
           ablow   := a * b + b * v * low
           abmed   := a * b + b * v * med
           abhig   := a * b + b * v * hig
           tolow   := ablow + cprime
           totmed   := abmed + cprime
           tohig   := abhig + cprime'

# Spécifier les paramètres du modèle
fit <- sem(model.2, data = data, missing = "listwise", estimator = "ML")

# Obtenir le sommaire des estimations du modèle
summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE))
```



Output 7.2. Sortie de l'analyse de médiation modérée du Listing 7.2. Les nombres en **magenta** sont les moyennes de chaque variable, les nombres en **vert** sont les coefficients de régression et leur valeur p , les nombres en **gris** sont les coefficients de régression standardisés, les nombres en **bleu** représentent la variance expliquée par la variable dépendante et médiatrice puis les nombres en **jaune** représentent l'estimée des paramètres créés et inclus dans le modèle.

```
> # Inspecter la base de données
> str(data)

'data.frame': 200 obs. of 4 variables:
 $ X: num  1.175 0.398 1.951 -0.905 -0.682 ...
 $ M: num  -0.027 0.364 0.795 -1.904 -1.355 ...
 $ V: num   0.984 0.101 -0.363 -0.739 -1.48 ...
 $ Y: num   0.611 0.527 -0.602 0.825 -0.433 ...

> # Analyses descriptives
> # Mesures de tendance centrale
> summary(data)
```

X	M	V	Y
Min. : -2.64100	Min. : -3.10800	Min. : -2.882000	Min. : -2.3380
1st Qu.: -0.51325	1st Qu.: -0.66700	1st Qu.: -0.637500	1st Qu.: -0.5915
Median : 0.02750	Median : 0.03450	Median : 0.067000	Median : 0.0300
Mean : 0.09559	Mean : 0.04847	Mean : 0.003265	Mean : -0.0241
3rd Qu.: 0.73275	3rd Qu.: 0.74650	3rd Qu.: 0.628000	3rd Qu.: 0.5148
Max. : 2.46500	Max. : 2.80600	Max. : 3.466000	Max. : 3.2440

```

      MV
Min. : -5.28065
1st Qu.: -0.43630
Median : -0.01862
Mean : -0.12463
3rd Qu.: 0.27398
Max. : 3.36807

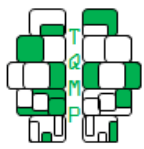
> # Covariances
> cov(data, y = NULL, use = "complete.obs", method = c("pearson"))
```

	X	M	V	Y	MV
X	0.92050441	0.2890840	-0.02611549	0.19034112	-0.0862648
M	0.28908399	1.1236741	-0.12541563	0.51540158	-0.2244063
V	-0.02611549	-0.1254156	1.07012701	0.07937768	0.1884572
Y	0.19034112	0.5154016	0.07937768	0.85470238	0.3131354
MV	-0.08626480	-0.2244063	0.18845723	0.31313541	1.1310276

```

> # Corrélations
> cor(data, y = NULL, use = "complete.obs", method = c("pearson"))
```

	X	M	V	Y	MV
X	1.00000000	0.2842437	-0.02631283	0.21459136	-0.08454425
M	0.28424375	1.0000000	-0.11437044	0.52591809	-0.19905740
V	-0.02631283	-0.1143704	1.00000000	0.08299909	0.17130046



```
Y    0.21459136  0.5259181  0.08299909 1.00000000  0.31848436
MV -0.08454425 -0.1990574  0.17130046 0.31848436  1.00000000
```

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

lavaan 0.6-7 ended normally after 17 iterations

```
Estimator           ML
Optimization method  NLMINB
Number of free parameters  7
Number of observations 200
```

Model Test User Model:

```
Test statistic      8.212
Degrees of freedom    2
P-value (Chi-square) 0.016
```

Model Test Baseline Model:

```
Test statistic      154.200
Degrees of freedom    7
P-value              0.000
```

Loglikelihood and Information Criteria:

```
Loglikelihood user model (H0)      -489.539
Loglikelihood unrestricted model (H1) -485.433
```

```
Akaike (AIC)          993.078
Bayesian (BIC)        1016.166
Sample-size adjusted Bayesian (BIC) 993.990
```

Root Mean Square Error of Approximation:

```
RMSEA              0.125
90 Percent confidence interval - lower 0.045
90 Percent confidence interval - upper 0.218
P-value RMSEA <= 0.05 0.059
```

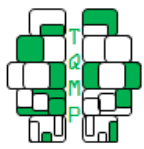
Standardized Root Mean Square Residual:

```
SRMR              0.070
```

Regressions:

		Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
Y ~							
X	(cprm)	0.080	0.052	1.556	0.120	0.080	0.080
M	(b)	0.521	0.047	11.180	0.000	0.521	0.569
V		0.071	0.046	1.533	0.125	0.071	0.076
MV	(v)	0.374	0.045	8.261	0.000	0.374	0.411
M ~							
X	(a)	0.314	0.075	4.193	0.000	0.314	0.284

Variances:



	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.Y	0.446	0.045	10.000	0.000	0.446	0.477
.M	1.028	0.103	10.000	0.000	1.028	0.919

R-Square:

	Estimate
Y	0.523
M	0.081

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
low	-1.000				-1.000	-1.000
med	0.000				0.000	0.000
hig	1.000				1.000	1.000
ablow	-0.031	0.046	-0.688	0.491	-0.031	-0.072
abmed	0.164	0.042	3.926	0.000	0.164	0.162
abhig	0.359	0.056	6.432	0.000	0.359	0.396
totlow	0.049	0.068	0.718	0.473	0.049	0.008
totmed	0.244	0.063	3.869	0.000	0.244	0.241
tothig	0.439	0.071	6.191	0.000	0.439	0.475

Annexe 8 : L'analyse de classes latentes (*Latent class analysis*)

Listing 8.1.1. Syntaxe pour réaliser une analyse de classes latentes avec deux classes latentes (modèle 1). Le fichier de données utilisé est `latent_class_data.csv`, les variables utilisées sont V1 à V5 puis la librairie nécessaire est `depmixS4`. Le fichier de syntaxe R Chapitre 16 – Analyse de classes latentes – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(depmixS4)

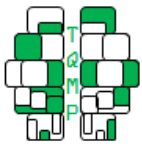
# Importer la base de données
data <- read.csv("latent_class_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Définir le modèle
# Modèle 1
model.1 <- depmix(list(V1 ~ 1, V2 ~ 1, V3 ~ 1, V4 ~ 1, V5 ~ 1), data = data,
  nstates = 2, family = list(gaussian(), gaussian(), gaussian(), gaussian(),
    gaussian()))

# Spécifier les paramètres du modèle
fit.1 <- fit(model.1)

# Obtenir le sommaire des estimations du modèle
fit.1
```

```
summary(fit.1)
```

Output 8.1.1. Sortie de l'analyse de classes latentes du Listing 8.1.1. Les nombres en **jaune** sont les indices d'ajustements.

```
> # Inspecter la base de données
> str(data)

'data.frame': 300 obs. of 8 variables:
 $ V1: num 1.5 2.24 2.64 5.06 1.48 ...
 $ V2: num 2.872 2.875 3.694 2.058 0.272 ...
 $ V3: num 4.05 2.41 3.42 3.53 1.75 ...
 $ V4: num 4.17 4.59 3.84 4.24 3.13 ...
 $ V5: num 4.5 2.56 2.29 2.33 3 ...
 $ V6: int 0 0 0 0 0 0 0 0 0 0 ...
 $ T1: num 0.586 -1.884 0.313 1.29 0.434 ...
 $ T2: num -1.005 -2.005 0.946 -0.407 -0.443 ...
```

```
> # Obtenir le sommaire des estimations du modèle
> fit.1
```

Convergence info: Log likelihood converged to within tol. (relative change)

'log Lik.' -2671.786 (df=23)

AIC: 5389.573

BIC: 5474.76

```
> summary(fit.1)
```

Initial state probabilities model

pr1 pr2

0 1

Transition matrix

toS1 toS2

fromS1 1.00 0.00

fromS2 0.01 0.99

Response parameters

Resp 1 : gaussian

Resp 2 : gaussian

Resp 3 : gaussian

Resp 4 : gaussian

Resp 5 : gaussian

Re1.(Intercept) Re1.sd Re2.(Intercept) Re2.sd

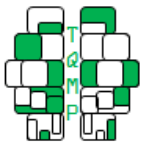
St1 -1.431 1.769 -1.506 1.846

St2 2.842 1.090 3.095 0.920

Re3.(Intercept) Re3.sd Re4.(Intercept) Re4.sd

St1 -1.518 1.703 -1.519 1.756

St2 2.985 0.944 3.174 0.933



```
Re5.(Intercept) Re5.sd
St1              -1.452  1.737
St2              3.010  0.852
```

Listing 8.1.2. Syntaxe pour réaliser une analyse de classes latentes avec trois classes latentes (modèle 2). Le fichier de données utilisé est `latent_class_data.csv`, les variables utilisées sont V1 à V5 puis la librairie nécessaire est `depmixS4`. Le fichier de syntaxe R Chapitre 16 – Analyse de classes latentes – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(depmixS4)

# Importer la base de données
data <- read.csv("latent_class_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Définir le modèle
# Modèle 2
model.2 <- depmix(list(V1 ~ 1, V2 ~ 1, V3 ~ 1, V4 ~ 1, V5 ~ 1), data = data,
  nstates = 3, family = list(gaussian(), gaussian(), gaussian(), gaussian(),
    gaussian()))

# Spécifier les paramètres du modèle
fit.2 <- fit(model.2)

# Obtenir le sommaire des estimations du modèle
fit.2

summary(fit.2)
```

Output 8.1.2. Sortie de l'analyse de classes latentes du Listing 8.1.2. Les nombres en **jaune** sont les indices d'ajustements.

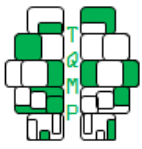
```
> # Obtenir le sommaire des estimations du modèle
> fit.2

Convergence info: Log likelihood converged to within tol. (relative change)
'log Lik.' -2119.951 (df=38)
AIC: 4315.902
BIC: 4456.646

> summary(fit.2)

Initial state probabilities model
pr1 pr2 pr3
  1   0   0

Transition matrix
  toS1 toS2 toS3
```



```
fromS1 0.99 0.01 0.00
fromS2 0.00 0.99 0.01
fromS3 0.00 0.00 1.00
```

```
Response parameters
Resp 1 : gaussian
Resp 2 : gaussian
Resp 3 : gaussian
Resp 4 : gaussian
Resp 5 : gaussian
```

	Re1.(Intercept)	Re1.sd	Re2.(Intercept)	Re2.sd
St1	2.842	1.090	3.095	0.920
St2	0.011	1.045	0.000	1.068
St3	-2.872	1.005	-3.013	1.066

	Re3.(Intercept)	Re3.sd	Re4.(Intercept)	Re4.sd
St1	2.985	0.944	3.174	0.933
St2	-0.107	0.878	-0.084	1.076
St3	-2.929	1.023	-2.954	0.942

	Re5.(Intercept)	Re5.sd
St1	3.010	0.852
St2	-0.035	1.067
St3	-2.868	0.940

Listing 8.1.3. Syntaxe pour réaliser une analyse de classes latentes avec quatre classes latentes (modèle 3). Le fichier de données utilisé est `latent_class_data.csv`, les variables utilisées sont V1 à V5 puis la librairie nécessaire est `depmixS4`. Le fichier de syntaxe R Chapitre 16 – Analyse de classes latentes – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(depmixS4)

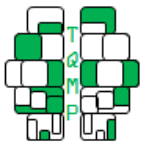
# Importer la base de données
data <- read.csv("latent_class_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Définir le modèle
# Modèle 3
model.3 <- depmix(list(V1 ~ 1, V2 ~ 1, V3 ~ 1, V4 ~ 1, V5 ~ 1), data = data,
  nstates = 4, family = list(gaussian(), gaussian(), gaussian(), gaussian(),
    gaussian()))

# Spécifier les paramètres du modèle
fit.3 <- fit(model.3)

# Obtenir le sommaire des estimations du modèle
```



```
fit.3
```

```
summary(fit.3)
```

Output 8.1.3. Sortie de l'analyse de classes latentes du Listing 8.1.3. Les nombres en **jaune** sont les indices d'ajustements.

```
> # Obtenir le sommaire des estimations du modèle
> fit.3
```

```
Convergence info: Log likelihood converged to within tol. (relative change)
```

```
'log Lik.' -2107.312 (df=55)
```

```
AIC: 4324.624
```

```
BIC: 4528.332
```

```
> summary(fit.3)
```

```
Initial state probabilities model
```

```
pr1 pr2 pr3 pr4
```

```
0 1 0 0
```

```
Transition matrix
```

	toS1	toS2	toS3	toS4
fromS1	0.635	0.00	0.316	0.05
fromS2	0.000	0.99	0.010	0.00
fromS3	0.092	0.00	0.908	0.00
fromS4	0.000	0.00	0.000	1.00

```
Response parameters
```

```
Resp 1 : gaussian
```

```
Resp 2 : gaussian
```

```
Resp 3 : gaussian
```

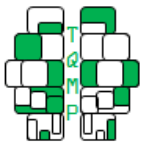
```
Resp 4 : gaussian
```

```
Resp 5 : gaussian
```

	Re1.(Intercept)	Re1.sd	Re2.(Intercept)	Re2.sd
St1	0.364	0.915	0.368	1.275
St2	2.842	1.090	3.095	0.920
St3	-0.078	1.056	-0.092	0.988
St4	-2.872	1.005	-3.013	1.066

	Re3.(Intercept)	Re3.sd	Re4.(Intercept)	Re4.sd
St1	-0.222	1.289	1.070	0.529
St2	2.985	0.944	3.174	0.933
St3	-0.078	0.736	-0.374	0.979
St4	-2.929	1.023	-2.954	0.942

	Re5.(Intercept)	Re5.sd
St1	0.084	1.144
St2	3.010	0.852
St3	-0.065	1.045
St4	-2.868	0.940



Listing 8.1.4. Syntaxe pour réaliser une analyse de classes latentes avec cinq classes latentes (modèle 4). Le fichier de données utilisé est `latent_class_data.csv`, les variables utilisées sont V1 à V5 puis la librairie nécessaire est `depmixS4`. Le fichier de syntaxe R Chapitre 16 – Analyse de classes latentes – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(depmixS4)

# Importer la base de données
data <- read.csv("latent_class_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Définir le modèle
# Modèle 4
model.4 <- depmix(list(V1 ~ 1, V2 ~ 1, V3 ~ 1, V4 ~ 1, V5 ~ 1), data = data,
  nstates = 5, family = list(gaussian(), gaussian(), gaussian(), gaussian(),
    gaussian()))

# Spécifier les paramètres du modèle
fit.4 <- fit(model.4)

# Obtenir le sommaire des estimations du modèle
fit.4

summary(fit.4)
```

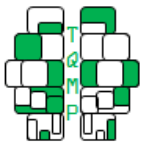
Output 8.1.4. Sortie de l'analyse de classes latentes du Listing 8.1.4. Les nombres en **jaune** sont les indices d'ajustements.

```
> # Obtenir le sommaire des estimations du modèle
> fit.4
Convergence info: Log likelihood converged to within tol. (relative change)
'log Lik.' -2090.671 (df=74)
AIC: 4329.341
BIC: 4603.421

> summary(fit.4)

Initial state probabilities model
pr1 pr2 pr3 pr4 pr5
  0   0   0   1   0

Transition matrix
      toS1 toS2 toS3 toS4 toS5
fromS1 0.000 1.000 0.000 0.00 0.000
fromS2 0.033 0.891 0.076 0.00 0.000
fromS3 0.409 0.000 0.511 0.00 0.081
fromS4 0.000 0.010 0.000 0.99 0.000
```



```
fromS5 0.000 0.000 0.000 0.00 1.000
```

```
Response parameters
```

```
Resp 1 : gaussian
```

```
Resp 2 : gaussian
```

```
Resp 3 : gaussian
```

```
Resp 4 : gaussian
```

```
Resp 5 : gaussian
```

	Re1.(Intercept)	Re1.sd	Re2.(Intercept)	Re2.sd
St1	-0.179	0.248	-1.112	0.945
St2	-0.047	1.081	-0.056	0.940
St3	0.505	0.975	1.058	1.021
St4	2.842	1.090	3.095	0.920
St5	-2.872	1.005	-3.013	1.066

	Re3.(Intercept)	Re3.sd	Re4.(Intercept)	Re4.sd
St1	-0.264	1.173	1.139	0.220
St2	-0.045	0.743	-0.367	1.001
St3	-0.408	1.293	0.980	0.525
St4	2.985	0.944	3.174	0.933
St5	-2.929	1.023	-2.954	0.942

	Re5.(Intercept)	Re5.sd
St1	1.163	0.538
St2	-0.140	1.033
St3	-0.106	1.081
St4	3.010	0.852
St5	-2.868	0.940

Listing 8.2. Syntaxe pour réaliser une analyse de classes latentes avec estimation automatique du meilleur modèle. Le fichier de données utilisé est `latent_class_data.csv`, les variables utilisées sont V1 à V5 puis la librairie nécessaire est `mclust`. Le fichier de syntaxe R Chapitre 16 – Analyse de classes latentes – `Input.R` se trouve dans le matériel accompagnateur.

```
# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(mclust)

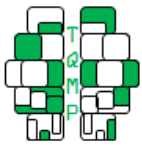
# Importer la base de données
data <- read.csv("latent_class_data.csv", header = TRUE, sep = ",")

# Modifier la base de données
# Sélectionner un sous-ensemble (V1-V5) de la base de données
data <- subset(data, select = c(V1, V2, V3, V4, V5))

# Inspecter la base de données
str(data)

# Définir le modèle
model <- Mclust(data)

# Identifier le modèle avec le meilleur ajustement
summary(model, parameters = TRUE)
```



Output 8.2. Sortie de l'analyse de classes latentes du Listing 8.2. Les nombres en **jaune** sont les indices d'ajustements.

```
> # Inspecter la base de données
> str(data)

'data.frame': 300 obs. of 5 variables:
 $ V1: num 1.5 2.24 2.64 5.06 1.48 ...
 $ V2: num 2.872 2.875 3.694 2.058 0.272 ...
 $ V3: num 4.05 2.41 3.42 3.53 1.75 ...
 $ V4: num 4.17 4.59 3.84 4.24 3.13 ...
 $ V5: num 4.5 2.56 2.29 2.33 3 ...

> # Identifier le modèle avec le meilleur ajustement
> summary(model, parameters = TRUE)

-----
Gaussian finite mixture model fitted by EM algorithm
-----

Mclust EII (spherical, equal volume) model with 3 components:

log-likelihood    n df    BIC          ICL
-2447.179    300 18   -4997.026  -4997.044

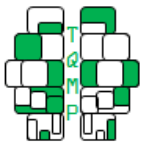
Clustering table:
 1  2  3
100 100 100

Mixing probabilities:
      1      2      3
0.3333196 0.3333426 0.3333377

Means:
      [,1]      [,2]      [,3]
V1 2.841989 0.0110098565 -2.872205
V2 3.094897 0.0004803656 -3.013191
V3 2.985079 -0.1067191193 -2.928486
V4 3.173913 -0.0835336843 -2.954258
V5 3.009642 -0.0351405865 -2.868139

Variances:
[, ,1]
      V1      V2      V3      V4      V5
V1 0.9857774 0.0000000 0.0000000 0.0000000 0.0000000
V2 0.0000000 0.9857774 0.0000000 0.0000000 0.0000000
V3 0.0000000 0.0000000 0.9857774 0.0000000 0.0000000
V4 0.0000000 0.0000000 0.0000000 0.9857774 0.0000000
V5 0.0000000 0.0000000 0.0000000 0.0000000 0.9857774

[, ,2]
      V1      V2      V3      V4      V5
V1 0.9857774 0.0000000 0.0000000 0.0000000 0.0000000
```



```
V2 0.0000000 0.9857774 0.0000000 0.0000000 0.0000000
V3 0.0000000 0.0000000 0.9857774 0.0000000 0.0000000
V4 0.0000000 0.0000000 0.0000000 0.9857774 0.0000000
V5 0.0000000 0.0000000 0.0000000 0.0000000 0.9857774
```

```
[,,3]
```

```
      V1      V2      V3      V4      V5
V1 0.9857774 0.0000000 0.0000000 0.0000000 0.0000000
V2 0.0000000 0.9857774 0.0000000 0.0000000 0.0000000
V3 0.0000000 0.0000000 0.9857774 0.0000000 0.0000000
V4 0.0000000 0.0000000 0.0000000 0.9857774 0.0000000
V5 0.0000000 0.0000000 0.0000000 0.0000000 0.9857774
```

Annexe 9 : L'analyse de modèles autorégressifs (*Autoregressive models analysis*)

Listing 9. Syntaxe pour réaliser une analyse de modèles autorégressifs. Le fichier de données utilisé est `autoreg_data.csv`, les variables utilisées sont V1 à V4 et la librairie R nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 17 – Analyse de modèles autorégressifs – Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("autoreg_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Analyses descriptives
# Mesures de tendance centrale
summary(data)

# Covariances
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

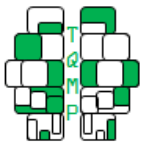
# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
model <- 'V4 ~ V3
         V3 ~ V2
         V2 ~ V1'

# Spécifier les paramètres du modèle
fit <- sem(model, data = data)

# Obtenir le sommaire des estimations du modèle
summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)

modificationindices(fit, standardized = TRUE)
```

Output 9. Sortie de l'analyse de modèles autorégressifs du Listing 9. Les nombres en **bleu** sont les moyennes de chaque variable, les nombres en **jaune** sont les indices d'ajustement, les nombres en **vert** sont les coefficients de régression et leur valeur p , les nombres en **gris** sont les coefficients de régression standardisés, les nombres en **rouge** représentent la variance expliquée par chacune des variables endogènes puis les nombres en **magenta** sont les indices de modifications.

```
> # Inspecter la base de données
> str(data)

'data.frame': 250 obs. of 4 variables:
 $ V1: num -1.605 0.592 0.46 -0.295 0.024 ...
 $ V2: num -0.895 0.79 0.186 0.952 -0.076 ...
 $ V3: num 0.057 -0.876 -0.021 1.121 0.283 ...
 $ V4: num 0.918 -0.368 -0.991 0.952 -0.337 -0.542 0.054 -0.604 0.464 -1.41 ...

> # Analyses descriptives
> # Mesures de tendance centrale
> summary (data)
```

V1	V2	V3	V4
Min. : -3.19300	Min. : -2.81300	Min. : -2.52200	Min. : -2.42200
1st Qu.: -0.79275	1st Qu.: -0.76725	1st Qu.: -0.60150	1st Qu.: -0.67600
Median : -0.01250	Median : 0.00850	Median : 0.02150	Median : 0.05950
Mean : -0.04114	Mean : -0.04758	Mean : 0.03013	Mean : 0.00172
3rd Qu.: 0.67600	3rd Qu.: 0.67125	3rd Qu.: 0.66350	3rd Qu.: 0.68850
Max. : 2.75300	Max. : 2.60600	Max. : 2.98800	Max. : 2.69100

```
> # Covariances
> cov(data, y = NULL, use = "complete.obs", method = c("pearson"))
```

	V1	V2	V3	V4
V1	1.0433273	0.5302352	0.1714306	0.1049174
V2	0.5302352	0.9714948	0.4479581	0.1816049
V3	0.1714306	0.4479581	0.8558123	0.4303606
V4	0.1049174	0.1816049	0.4303606	0.9392145

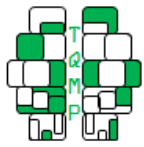
```
> # Corrélations
> cor(data, y = NULL, use = "complete.obs", method = c("pearson"))
```

	V1	V2	V3	V4
V1	1.0000000	0.5266693	0.1814215	0.1059876
V2	0.5266693	1.0000000	0.4912787	0.1901188
V3	0.1814215	0.4912787	1.0000000	0.4800216
V4	0.1059876	0.1901188	0.4800216	1.0000000

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

lavaan 0.6-7 ended normally after 11 iterations

Estimator	ML
Optimization method	NLMINB



Number of free parameters	6
Number of observations	250

Model Test User Model:

Test statistic	4.574
Degrees of freedom	3
P-value (Chi-square)	0.206

Model Test Baseline Model:

Test statistic	220.325
Degrees of freedom	6
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.993
Tucker-Lewis Index (TLI)	0.985

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-923.908
Loglikelihood unrestricted model (H1)	-921.622

Akaike (AIC)	1859.817
Bayesian (BIC)	1880.946
Sample-size adjusted Bayesian (BIC)	1861.925

Root Mean Square Error of Approximation:

RMSEA	0.046
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.124
P-value RMSEA <= 0.05	0.437

Standardized Root Mean Square Residual:

SRMR	0.029
------	-------

Regressions:

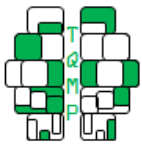
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
V4 ~						
V3	0.503	0.058	8.652	0.000	0.503	0.480
V3 ~						
V2	0.461	0.052	8.918	0.000	0.461	0.491
V2 ~						
V1	0.508	0.052	9.796	0.000	0.508	0.527

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.V4	0.720	0.064	11.180	0.000	0.720	0.770
.V3	0.647	0.058	11.180	0.000	0.647	0.759
.V2	0.699	0.063	11.180	0.000	0.699	0.723

R-Square:

Estimate



V4	0.230
V3	0.241
V2	0.277

```
> modificationindices(fit, standardized = TRUE)
```

```
[1] lhs      op      rhs      mi      epc      sepc.lv  sepc.all sepc.nox
<0 rows> (or 0-length row.names)
```

Annexe 10 : L'analyse de modèles autorégressifs croisés (*Crossed-lagged panel models*)

Listing 10. Syntaxe pour réaliser une analyse de modèles autorégressifs croisés. Le fichier de données utilisé est `cross_lagged_data`. Les variables utilisées sont X1, X2, Y1 et Y2 puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 18 – Analyse de modèles autorégressifs croisés – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("cross_lagged_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Définir le modèle
model <- 'X2 ~ X1 + Y1
         Y2 ~ X1 + Y1'

# Spécifier les paramètres du modèle
fit <- sem(model, data = data)

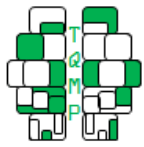
# Obtenir le sommaire des estimations du modèle
summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 10. Sortie de l'analyse de modèles autorégressifs croisés du Listing 10. Les nombres en **jaune** sont les coefficients de régression et leur valeur p , les nombres en **vert** sont les coefficients de régression standardisés puis les nombres en **bleu** représentent la variance expliquée par les deux variables indépendantes.

```
> # Inspecter la base de données
> str(data)

'data.frame': 150 obs. of 4 variables:
 $ X1: num 0.539 0.452 0.348 -2.41 0.194 ...
 $ X2: num 1.579 0.093 -0.312 -0.837 -1.683 ...
 $ Y1: num 1.335 0.144 -0.407 0.135 -2.333 ...
 $ Y2: num 1.091 1.354 -0.814 -1.466 -1.707 ...

> # Obtenir le sommaire des estimations du modèle
> summary(fit, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```



lavaan 0.6-7 ended normally after 36 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	7
Number of observations	150

Model Test User Model:

Test statistic	0.000
Degrees of freedom	0

Model Test Baseline Model:

Test statistic	458.983
Degrees of freedom	5
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	1.000
Tucker-Lewis Index (TLI)	1.000

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-211.328
Loglikelihood unrestricted model (H1)	-211.328

Akaike (AIC)	436.657
Bayesian (BIC)	457.731
Sample-size adjusted Bayesian (BIC)	435.578

Root Mean Square Error of Approximation:

RMSEA	0.000
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.000
P-value RMSEA <= 0.05	NA

Standardized Root Mean Square Residual:

SRMR	0.000
------	-------

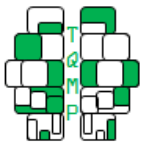
Regressions:

		Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
X2 ~							
X1	(β_1)	0.393	0.027	14.291	0.000	0.393	0.391
Y1	(β_2)	0.876	0.028	31.001	0.000	0.876	0.847
Y2 ~							
X1	(β_3)	0.401	0.054	7.502	0.000	0.401	0.397
Y1	(β_4)	0.665	0.055	12.074	0.000	0.665	0.639

Covariances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.X2 ~~						
.Y2	-0.013	0.020	-0.672	0.502	-0.013	-0.055

Variances:



	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.X2	0.123	0.014	8.660	0.000	0.123	0.112
.Y2	0.467	0.054	8.660	0.000	0.467	0.420

R-Square:

	Estimate
X2	0.888
Y2	0.580

Annexe 11 : L'analyse de trajectoire latente (*Latent growth analysis*)

Listing 11.1. Syntaxe pour réaliser une analyse de trajectoire latente avec un modèle linéaire sans covariable (modèle 1). Le fichier de données utilisé est `growth_data_1.csv`, les variables utilisées sont T1 à T4, `cov1t1` à `cov1t4` et `cov2` puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 19 – Analyse de trajectoire latente – Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("growth_data_1.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Analyses descriptives
# Mesures de tendance centrale
summary(data)

# Covariances
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

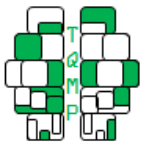
# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
# Modèle 1 (modèle linéaire sans covariables)
model.1 <- 'i =~ 1*T1 + 1*T2 + 1*T3 + 1*T4
           s =~ 0*T1 + 1*T2 + 2*T3 + 3*T4'

# Spécifier les paramètres du modèle
fit.1 <- growth(model.1, data = data)

# Obtenir le sommaire des estimations du modèle
summary(fit.1, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 11.1. Sortie de l'analyse de trajectoire latente du Listing 11.1. Les nombres en magenta sont les moyennes de chaque variable, les nombres en jaune sont les indices d'ajustement, les nombres en vert sont les scores latents pour la pente et l'ordonnée à l'origine puis les nombres en bleu représentent les variances expliquées par chacun des temps de mesure.



```
> # Examiner la base de données
> str(data)
```

```
'data.frame': 600 obs. of 9 variables:
 $ T1      : num  3.92  3.95  3.08  2.12  5.37 ...
 $ T2      : num  2.92  2.85  2.14  2.48  4.71 ...
 $ T3      : num  2.74  2.65  1.32  1.63  3.6 ...
 $ T4      : num  1.372 1.879 0.654 0.827 2.249 ...
 $ cov1t1 : num -0.363 -0.332 0.385 -1.732 1.084 ...
 $ cov1t2 : num  0.035 -1.277 -0.285 1 1.778 ...
 $ cov1t3 : num -0.439 0.06 -0.547 -0.342 -0.049 ...
 $ cov1t4 : num -1.289 1.392 -0.511 0.252 -1.548 ...
 $ cov2    : num -0.047 -0.325 0.231 -0.347 1.429 ...
```

```
> # Analyses descriptives
> # Mesures de tendance centrale
> summary(data)
```

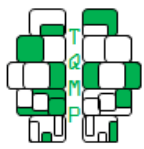
T1	T2	T3	T4
Min. : 0.629	Min. : -0.088	Min. : -1.376	Min. : -2.3730
1st Qu.: 3.207	1st Qu.: 2.368	1st Qu.: 1.575	1st Qu.: 0.8748
Median : 3.925	Median : 3.217	Median : 2.426	Median : 1.5810
Mean : 4.004	Mean : 3.200	Mean : 2.398	Mean : 1.6005
3rd Qu.: 4.776	3rd Qu.: 3.980	3rd Qu.: 3.123	3rd Qu.: 2.4057
Max. : 7.249	Max. : 6.358	Max. : 5.294	Max. : 4.8120

cov1t1	cov1t2	cov1t3	cov1t4
Min. : -3.05900	Min. : -2.91900	Min. : -3.168000	Min. : -2.91600
1st Qu.: -0.63550	1st Qu.: -0.57025	1st Qu.: -0.652500	1st Qu.: -0.77875
Median : -0.02600	Median : 0.07250	Median : 0.029000	Median : -0.12750
Mean : 0.01733	Mean : 0.05543	Mean : -0.001295	Mean : -0.08997
3rd Qu.: 0.69150	3rd Qu.: 0.70850	3rd Qu.: 0.641500	3rd Qu.: 0.59075
Max. : 2.53500	Max. : 2.48300	Max. : 2.908000	Max. : 3.13100

```
cov2
Min. : -1.78100
1st Qu.: -0.36800
Median : -0.02950
Mean : -0.01372
3rd Qu.: 0.32825
Max. : 1.77300
```

```
> # Covariances
> cov(data, y = NULL, use = "complete.obs", method = c("pearson"))
```

	T1	T2	T3	T4
T1	1.27259694	1.020040623	1.014207385	1.072080130
T2	1.02004062	1.181235743	0.990416657	1.042491860
T3	1.01420738	0.990416657	1.189115247	1.059673948
T4	1.07208013	1.042491860	1.059673948	1.289384611
cov1t1	0.48472760	0.031910118	-0.004691770	0.012106735
cov1t2	-0.04599615	0.302536249	-0.064887180	-0.071265341



```
cov1t3 -0.01636824 -0.053982151 0.277312807 -0.008466833
cov1t4 0.08551887 0.104805434 0.077431183 0.303639792
cov2 -0.00399046 -0.005202368 0.007593475 0.000321752
```

```
      cov1t1      cov1t2      cov1t3      cov1t4
T1      0.484727602 -0.045996151 -0.016368239 0.0855188714
T2      0.031910118 0.302536249 -0.053982151 0.1048054336
T3     -0.004691770 -0.064887180 0.277312807 0.0774311825
T4      0.012106735 -0.071265341 -0.008466833 0.3036397924
cov1t1 0.951486221 0.017365487 -0.007860582 -0.0098479632
cov1t2 0.017365487 0.872372445 -0.064373764 0.0011207418
cov1t3 -0.007860582 -0.064373764 0.917015571 -0.0209882482
cov1t4 -0.009847963 0.001120742 -0.020988248 1.0425677420
cov2 -0.002258495 -0.002593438 0.013138338 0.0008403123
```

```
      cov2
T1     -0.0039904597
T2     -0.0052023677
T3      0.0075934751
T4      0.0003217520
cov1t1 -0.0022584949
cov1t2 -0.0025934376
cov1t3 0.0131383384
cov1t4 0.0008403123
cov2    0.2532583199
```

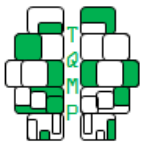
```
> # Corrélations
```

```
> cor(data, y = NULL, use = "complete.obs", method = c("pearson"))
```

```
      T1      T2      T3      T4
T1      1.000000000 0.831962482 0.824459560 0.8369327630
T2      0.831962482 1.000000000 0.835675505 0.8447207091
T3      0.824459560 0.835675505 1.000000000 0.8557936160
T4      0.836932763 0.844720709 0.855793616 1.0000000000
cov1t1 0.440505223 0.030099471 -0.004410863 0.0109303621
cov1t2 -0.043654105 0.298028748 -0.063708291 -0.0671948758
cov1t3 -0.015151937 -0.051867288 0.265564243 -0.0077864845
cov1t4 0.074244516 0.094441607 0.069542737 0.2618878215
cov2 -0.007029034 -0.009511541 0.013837153 0.0005630516
```

```
      cov1t1      cov1t2      cov1t3      cov1t4
T1      0.440505223 -0.043654105 -0.015151937 0.074244516
T2      0.030099471 0.298028748 -0.051867288 0.094441607
T3     -0.004410863 -0.063708291 0.265564243 0.069542737
T4      0.010930362 -0.067194876 -0.007786484 0.261887821
cov1t1 1.000000000 0.019060520 -0.008415213 -0.009887649
cov1t2 0.019060520 1.000000000 -0.071972996 0.001175175
cov1t3 -0.008415213 -0.071972996 1.000000000 -0.021465237
cov1t4 -0.009887649 0.001175175 -0.021465237 1.000000000
cov2 -0.004600828 -0.005517508 0.027262785 0.001635335
```

```
      cov2
```



```
T1      -0.0070290338
T2      -0.0095115412
T3       0.0138371529
T4       0.0005630516
cov1t1  -0.0046008281
cov1t2  -0.0055175080
cov1t3   0.0272627855
cov1t4   0.0016353349
cov2     1.0000000000
```

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.1, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

lavaan 0.6-7 ended normally after 47 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	9
Number of observations	600

Model Test User Model:

Test statistic	6.063
Degrees of freedom	5
P-value (Chi-square)	0.300

Model Test Baseline Model:

Test statistic	2533.232
Degrees of freedom	6
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	1.000
Tucker-Lewis Index (TLI)	0.999

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-2390.365
Loglikelihood unrestricted model (H1)	-2387.333

Akaike (AIC)	4798.730
Bayesian (BIC)	4838.302
Sample-size adjusted Bayesian (BIC)	4809.730

Root Mean Square Error of Approximation:

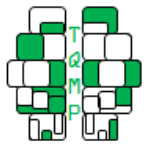
RMSEA	0.019
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.062
P-value RMSEA <= 0.05	0.856

Standardized Root Mean Square Residual:

SRMR	0.022
------	-------

Latent Variables:

Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
----------	---------	---------	---------	--------	---------



i =~				
T1	1.000		1.001	0.903
T2	1.000		1.001	0.909
T3	1.000		1.001	0.902
T4	1.000		1.001	0.895
s =~				
T1	0.000		0.000	0.000
T2	1.000		0.057	0.052
T3	2.000		0.114	0.103
T4	3.000		0.171	0.153

Covariances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
i ~~						
s	0.008	0.010	0.735	0.462	0.135	0.135

Intercepts:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.T1	0.000				0.000	0.000
.T2	0.000				0.000	0.000
.T3	0.000				0.000	0.000
.T4	0.000				0.000	0.000
i	4.002	0.044	91.396	0.000	3.999	3.999
s	-0.801	0.008	-95.121	0.000	-14.044	-14.044

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.T1	0.226	0.022	10.283	0.000	0.226	0.184
.T2	0.192	0.015	12.953	0.000	0.192	0.159
.T3	0.187	0.015	12.674	0.000	0.187	0.152
.T4	0.172	0.020	8.809	0.000	0.172	0.138
i	1.001	0.067	14.901	0.000	1.000	1.000
s	0.003	0.004	0.835	0.404	1.000	1.000

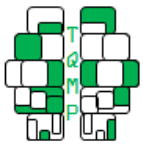
R-Square:

	Estimate
T1	0.816
T2	0.841
T3	0.848
T4	0.862

Listing 11.2. Syntaxe pour réaliser une analyse de trajectoire latente avec un modèle quadratique sans covariable (modèle 2). Le fichier de données utilisé est `growth_data_1.csv`, les variables utilisées sont T1 à T4 puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 19 – Analyse de trajectoire latente – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)
```



```
# Importer la base de données
data <- read.csv("growth_data_1.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Analyses descriptives
# Mesures de tendance centrale
summary (data)

# Covariances
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
# Modèle 2 (modèle quadratique sans covariable)
model.2 <- 'i =~ 1*T1 + 1*T2 + 1*T3 + 1*T4
           s =~ 0*T1 + 1*T2 + 2*T3 + 3*T4
           q =~ 0*T1 + 1*T2 + 4*T3 + 9*T4'

# Spécifier les paramètres du modèle
fit.2 <- growth(model.2, data = data)

# Obtenir le sommaire des estimations du modèle
summary(fit.2, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 11.2. Sortie de l'analyse de trajectoire latente du Listing 11.2. Les nombres en **jaune** sont les indices d'ajustement, les nombres en **vert** sont les coefficients de régression pour l'effet quadratique, puis les nombres en **bleu** représentent les variances expliquées par chacun des temps de mesure.

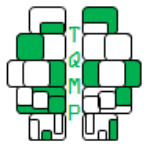
```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.2, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

```
lavaan 0.6-7 ended normally after 65 iterations
  Estimator                      ML
  Optimization method            NLMINB
  Number of free parameters      13
  Number of observations         600

Model Test User Model:
  Test statistic                  0.001
  Degrees of freedom              1
  P-value (Chi-square)           0.969

Model Test Baseline Model:
  Test statistic                  2533.232
  Degrees of freedom              6
  P-value                        0.000

User Model versus Baseline Model:
  Comparative Fit Index (CFI)     1.000
```



Tucker-Lewis Index (TLI) 1.002

Loglikelihood and Information Criteria:

Loglikelihood user model (H0) -2387.334

Loglikelihood unrestricted model (H1) -2387.333

Akaike (AIC) 4800.668

Bayesian (BIC) 4857.828

Sample-size adjusted Bayesian (BIC) 4816.557

Root Mean Square Error of Approximation:

RMSEA 0.000

90 Percent confidence interval - lower 0.000

90 Percent confidence interval - upper 0.000

P-value RMSEA ≤ 0.05 0.985

Standardized Root Mean Square Residual:

SRMR 0.000

Latent Variables:

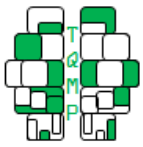
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
i =~						
T1	1.000				1.043	0.925
T2	1.000				1.043	0.960
T3	1.000				1.043	0.957
T4	1.000				1.043	0.919
s =~						
T1	0.000				0.000	0.000
T2	1.000				0.234	0.216
T3	2.000				0.468	0.430
T4	3.000				0.702	0.619
q =~						
T1	0.000				0.000	0.000
T2	1.000				0.058	0.053
T3	4.000				0.232	0.213
T4	9.000				0.522	0.460

Covariances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
i ~~						
s	-0.101	0.099	-1.025	0.305	-0.415	-0.415
q	0.032	0.024	1.310	0.190	0.526	0.526
s ~~						
q	-0.013	0.020	-0.644	0.520	-0.950	-0.950

Intercepts:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.T1	0.000				0.000	0.000
.T2	0.000				0.000	0.000
.T3	0.000				0.000	0.000
.T4	0.000				0.000	0.000
i	4.004	0.046	87.315	0.000	3.839	3.839



s	-0.806	0.029	-27.758	0.000	-3.441	-3.441
q	0.002	0.009	0.167	0.868	0.026	0.026

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.T1	0.183	0.090	2.035	0.042	0.183	0.144
.T2	0.198	0.030	6.634	0.000	0.198	0.168
.T3	0.183	0.030	6.153	0.000	0.183	0.154
.T4	0.165	0.090	1.833	0.067	0.165	0.129
i	1.088	0.109	9.975	0.000	1.000	1.000
s	0.055	0.097	0.564	0.573	1.000	1.000
q	0.003	0.005	0.712	0.477	1.000	1.000

R-Square:

	Estimate
T1	0.856
T2	0.832
T3	0.846
T4	0.871

Listing 11.3. Syntaxe pour réaliser une analyse de trajectoire latente avec un modèle linéaire avec covariable stable temporellement (modèle 3). Le fichier de données utilisé est `growth_data_1.csv`, les variables utilisées sont T1 à T4 et `cov2` puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 19 – Analyse de trajectoire latente – `Input.R` se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("growth_data_1.csv", header = TRUE, sep = ",")

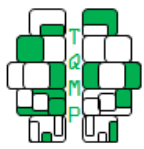
# Inspecter la base de données
str(data)

# Analyses descriptives
# Mesures de tendance centrale
summary(data)

# Covariances
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
# Modèle 3 (modèle linéaire avec covariable stable temporellement)
model.3 <- 'i =~ 1*T1 + 1*T2 + 1*T3 + 1*T4
            s =~ 0*T1 + 1*T2 + 2*T3 + 3*T4
            i ~ cov2
            s ~ cov2'
```



```
# Spécifier les paramètres du modèle
fit.3 <- growth(model.3, data = data)
```

```
# Obtenir le sommaire des estimations du modèle
summary(fit.3, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 11.3. Sortie de l'analyse de trajectoire latente du Listing 11.3. Les nombres en **jaune** sont les indices d'ajustement, les nombres en **vert** sont les coefficients de régression et les scores latents pour la pente et l'ordonnée à l'origine puis les nombres en **bleu** représentent les variances expliquées par chacun des temps de mesure, incluant l'effet de la covariable.

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.3, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

```
lavaan 0.6-7 ended normally after 47 iterations
  Estimator                      ML
  Optimization method            NLMINB
  Number of free parameters      11
  Number of observations         600

Model Test User Model:
  Test statistic                 6.885
  Degrees of freedom             7
  P-value (Chi-square)          0.441

Model Test Baseline Model:
  Test statistic                2534.401
  Degrees of freedom            10
  P-value                       0.000

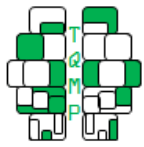
User Model versus Baseline Model:
  Comparative Fit Index (CFI)    1.000
  Tucker-Lewis Index (TLI)      1.000

Loglikelihood and Information Criteria:
  Loglikelihood user model (H0)  -2390.191
  Loglikelihood unrestricted model (H1) -2386.749

  Akaike (AIC)                  4802.383
  Bayesian (BIC)                 4850.749
  Sample-size adjusted Bayesian (BIC) 4815.827

Root Mean Square Error of Approximation:
  RMSEA                         0.000
  90 Percent confidence interval - lower 0.000
  90 Percent confidence interval - upper 0.050
  P-value RMSEA <= 0.05         0.952

Standardized Root Mean Square Residual:
  SRMR                          0.019
```



Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
i =~						
T1	1.000				1.001	0.903
T2	1.000				1.001	0.909
T3	1.000				1.001	0.902
T4	1.000				1.001	0.895
s =~						
T1	0.000				0.000	0.000
T2	1.000				0.057	0.052
T3	2.000				0.113	0.102
T4	3.000				0.170	0.152

Regressions:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
i ~						
cov2	-0.016	0.087	-0.186	0.852	-0.016	-0.008
s ~						
cov2	0.010	0.017	0.588	0.557	0.174	0.087

Covariances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.i ~~						
.s	0.008	0.010	0.742	0.458	0.138	0.138

Intercepts:

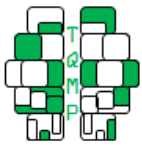
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.T1	0.000				0.000	0.000
.T2	0.000				0.000	0.000
.T3	0.000				0.000	0.000
.T4	0.000				0.000	0.000
.i	4.002	0.044	91.360	0.000	3.999	3.999
.s	-0.801	0.008	-95.095	0.000	-14.124	-14.124

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.T1	0.226	0.022	10.291	0.000	0.226	0.184
.T2	0.192	0.015	12.951	0.000	0.192	0.159
.T3	0.187	0.015	12.672	0.000	0.187	0.152
.T4	0.173	0.020	8.823	0.000	0.173	0.138
.i	1.001	0.067	14.900	0.000	1.000	1.000
.s	0.003	0.004	0.819	0.413	0.992	0.992

R-Square:

	Estimate
T1	0.816
T2	0.841
T3	0.848
T4	0.862
i	0.000
s	0.008



Listing 11.4. Syntaxe pour réaliser une analyse de trajectoire latente avec un modèle linéaire avec covariable instable temporellement (modèle 4). Le fichier de données utilisé est `growth_data_1.csv`, les variables utilisées sont T1 à T4 et `cov1t1` à `cov1t4` puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 19 – Analyse de trajectoire latente – Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("growth_data_1.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Analyses descriptives
# Mesures de tendance centrale
summary(data)

# Covariances
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
# Modèle 4 (modèle linéaire avec covariable instable temporellement)
model.4 <- 'i =~ 1*T1 + 1*T2 + 1*T3 + 1*T4
            s =~ 0*T1 + 1*T2 + 2*T3 + 3*T4
            T1 ~ cov1t1
            T2 ~ cov1t2
            T3 ~ cov1t3
            T4 ~ cov1t4
            s ~~ 0*s
            s ~~ 0*i'

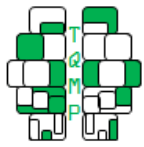
# Spécifier les paramètres du modèle
fit.4 <- growth(model.4, data = data)

# Obtenir le sommaire des estimations du modèle
summary(fit.4, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

Output 11.4. Sortie de l'analyse de trajectoire latente du Listing 11.4. Les nombres en **jaune** sont les indices d'ajustement, les nombres en **vert** sont les coefficients de régression et les scores latents pour la pente et l'ordonnée à l'origine, le nombre en **magenta** confirme que la variance de la pente a été fixée à 0 puis les nombres en **bleu** représentent les variances expliquées par chacun des temps de mesure.

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.4, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)

lavaan 0.6-7 ended normally after 82 iterations
```



Estimator	ML
Optimization method	NLMINB
Number of free parameters	11
Number of observations	600

Model Test User Model:

Test statistic	27.130
Degrees of freedom	19
P-value (Chi-square)	0.102

Model Test Baseline Model:

Test statistic	4903.163
Degrees of freedom	22
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.998
Tucker-Lewis Index (TLI)	0.998

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-1215.933
Loglikelihood unrestricted model (H1)	-1202.368

Akaike (AIC)	2453.866
Bayesian (BIC)	2502.233
Sample-size adjusted Bayesian (BIC)	2467.311

Root Mean Square Error of Approximation:

RMSEA	0.027
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.048
P-value RMSEA ≤ 0.05	0.967

Standardized Root Mean Square Residual:

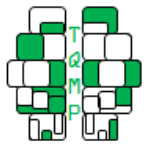
SRMR	0.034
------	-------

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
i =~						
T1	1.000				1.011	0.902
T2	1.000				1.011	0.917
T3	1.000				1.011	0.923
T4	1.000				1.011	0.922
s =~						
T1	0.000				0.000	0.000
T2	1.000				0.000	0.000
T3	2.000				0.000	0.000
T4	3.000				0.000	0.000

Regressions:

Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
----------	---------	---------	---------	--------	---------



T1 ~ cov1t1	0.494	0.007	70.563	0.000	0.494	0.430
T2 ~ cov1t2	0.408	0.010	40.758	0.000	0.408	0.345
T3 ~ cov1t3	0.317	0.013	24.770	0.000	0.317	0.276
T4 ~ cov1t4	0.203	0.015	13.586	0.000	0.203	0.189

Covariances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
i ~~ s	0.000				NaN	NaN

Intercepts:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.T1	0.000				0.000	0.000
.T2	0.000				0.000	0.000
.T3	0.000				0.000	0.000
.T4	0.000				0.000	0.000
i	3.994	0.041	96.629	0.000	3.950	3.950
s	-0.798	0.004	-214.328	0.000	-Inf	-Inf

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
s	0.000				NaN	NaN
.T1	0.003	0.002	1.462	0.144	0.003	0.002
.T2	0.050	0.004	14.203	0.000	0.050	0.041
.T3	0.087	0.005	15.992	0.000	0.087	0.073
.T4	0.137	0.008	16.653	0.000	0.137	0.114
i	1.022	0.059	17.271	0.000	1.000	1.000

R-Square:

	Estimate
T1	0.998
T2	0.959
T3	0.927
T4	0.886

Listing 11.5. Syntaxe pour réaliser une analyse de trajectoire latente pour un modèle linéaire avec trajectoires latentes parallèles (modèle 5). Le fichier de données utilisé est `growth_data_2.csv`, les variables utilisées sont X1 à X4 et Y1 à Y4 puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 19 – Analyse de trajectoire latente – Input .R se trouve dans le matériel accompagnateur.

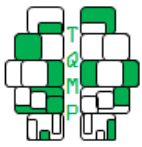
```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("growth_data_2.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Analyses descriptives
```



```
# Mesures de tendance centrale
summary (data)

# Covariances
cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Corrélations
cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

# Définir le modèle
# Modèle 5 (modèle linéaire avec trajectoires latentes parallèles)
model.5 <- 'iX =~ 1*X1 + 1*X2 + 1*X3 + 1*X4
            sX =~ 0*X1 + 1*X2 + 2*X3 + 3*X4
            iY =~ 1*Y1 + 1*Y2 + 1*Y3 + 1*Y4
            sY =~ 0*Y1 + 1*Y2 + 2*Y3 + 3*Y4
            sX ~~ 0.01*sX'

# Spécifier les paramètres du modèle
Fit.5 <- growth(model.5, data = data)

# Obtenir le sommaire des estimations du modèle
summary(fit.5, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

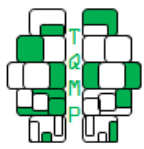
Output 11.5. Sortie de l'analyse de trajectoire latente du Listing 11.5. Les nombres en **rouge** sont les moyennes de chaque variable, les nombres en **jaune** sont les indices d'ajustement, les nombres en **gris** sont les corrélations entre les variables latentes, les nombres en **vert** sont les scores latents pour la pente et l'ordonnée à l'origine, le nombre en **magenta** confirme que la variance de sX a été fixée à 0.01 puis les nombres en **bleu** représentent les variances expliquées par chacun des temps de mesure.

```
> # Inspecter la base de données
> str(data)

' data.frame': 600 obs. of 8 variables:
 $ X1: num 4.61 3.3 4.25 3.32 2.66 ...
 $ X2: num 0.731 2.341 2.368 4.73 2.066 ...
 $ X3: num 4.281 2.335 1.534 3.714 -0.302 ...
 $ X4: num 1.869 0.253 2.584 1.155 -0.286 ...
 $ Y1: num 0.338 -0.268 1.574 -0.05 -0.319 ...
 $ Y2: num -0.667 0.71 2.437 5.057 -3.355 ...
 $ Y3: num 0.501 1.472 1.614 3.506 -2.753 ...
 $ Y4: num 1.219 0.469 3.722 0.415 -1.31 ...

> # Analyses descriptives
> # Mesures de tendance centrale
> summary(data)
```

X1	X2	X3	X4
Min. :0.999	Min. :-1.787	Min. :-2.191	Min. :-1.5920
1st Qu.:3.280	1st Qu.: 2.073	1st Qu.: 1.363	1st Qu.: 0.6208
Median :3.959	Median : 3.127	Median : 2.372	Median : 1.5400
Mean :3.964	Mean : 3.168	Mean : 2.389	Mean : 1.5496
3rd Qu.:4.638	3rd Qu.: 4.328	3rd Qu.: 3.297	3rd Qu.: 2.3910
Max. :7.038	Max. : 7.853	Max. : 6.612	Max. : 5.2940



Y1	Y2	Y3	Y4
Min. : -2.64800	Min. : -6.2950	Min. : -6.714	Min. : -3.554
1st Qu.: -0.71150	1st Qu.: -1.1540	1st Qu.: -0.452	1st Qu.: 0.134
Median : 0.02250	Median : 0.5685	Median : 0.811	Median : 1.351
Mean : 0.05395	Mean : 0.5849	Mean : 0.865	Mean : 1.236
3rd Qu.: 0.73900	3rd Qu.: 2.5158	3rd Qu.: 2.191	3rd Qu.: 2.310
Max. : 3.00700	Max. : 7.3190	Max. : 5.784	Max. : 6.725

> # Covariances

> cov(data, y = NULL, use = "complete.obs", method = c("pearson"))

	X1	X2	X3	X4
X1	0.951431833	0.81247671	0.95334631	0.98883617
X2	0.812476708	2.79245855	0.87742121	0.80979616
X3	0.953346314	0.87742121	2.01208734	1.05168879
X4	0.988836169	0.80979616	1.05168879	1.54466467
Y1	0.006575101	-0.07890903	0.02214961	-0.01818924
Y2	-0.006680588	-0.06978222	0.16938648	0.03964087
Y3	0.162384778	-0.01773018	0.18319654	0.09869335
Y4	-0.059802685	-0.11210840	-0.04830526	-0.10027502

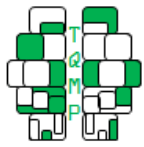
	Y1	Y2	Y3	Y4
X1	0.006575101	-0.006680588	0.16238478	-0.05980268
X2	-0.078909029	-0.069782216	-0.01773018	-0.11210840
X3	0.022149606	0.169386484	0.18319654	-0.04830526
X4	-0.018189237	0.039640869	0.09869335	-0.10027502
Y1	1.094391338	1.189802232	1.08825280	1.16909863
Y2	1.189802232	6.528743881	1.26322109	1.38502282
Y3	1.088252798	1.263221090	3.50952042	1.22438757
Y4	1.169098635	1.385022824	1.22438757	2.78913262

> # Corrélations

> cor(data, y = NULL, use = "complete.obs", method = c("pearson"))

	X1	X2	X3	X4
X1	1.00000000	0.498458163	0.68903049	0.81567734
X2	0.49845816	1.00000000	0.37016158	0.38991078
X3	0.68903049	0.370161585	1.00000000	0.59654983
X4	0.81567734	0.389910778	0.59654983	1.00000000
Y1	0.00644358	-0.045138488	0.01492644	-0.01398979
Y2	-0.00268047	-0.016343180	0.04673481	0.01248278
Y3	0.08886538	-0.005663642	0.06893979	0.04238839
Y4	-0.03671109	-0.040170773	-0.02039090	-0.04831045

	Y1	Y2	Y3	Y4
X1	0.00644358	-0.00268047	0.088865377	-0.03671109
X2	-0.04513849	-0.01634318	-0.005663642	-0.04017077
X3	0.01492644	0.04673481	0.068939791	-0.02039090
X4	-0.01398979	0.01248278	0.042388388	-0.04831045
Y1	1.00000000	0.44511644	0.555289654	0.66916042
Y2	0.44511644	1.00000000	0.263900775	0.32456937
Y3	0.55528965	0.26390077	1.00000000	0.39134559



Y4 0.66916042 0.32456937 0.391345588 1.00000000

> # Obtenir le sommaire des estimations du modèle

> summary(fit.5, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)

lavaan 0.6-7 ended normally after 55 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	21
Number of observations	600

Model Test User Model:

Test statistic	30.553
Degrees of freedom	23
P-value (Chi-square)	0.134

Model Test Baseline Model:

Test statistic	1949.397
Degrees of freedom	28
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.996
Tucker-Lewis Index (TLI)	0.995

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-7755.101
Loglikelihood unrestricted model (H1)	-7739.824

Akaike (AIC)	15552.202
Bayesian (BIC)	15644.537
Sample-size adjusted Bayesian (BIC)	15577.868

Root Mean Square Error of Approximation:

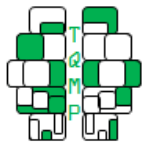
RMSEA	0.023
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.043
P-value RMSEA <= 0.05	0.989

Standardized Root Mean Square Residual:

SRMR	0.039
------	-------

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
iX =~						
X1	1.000				0.972	0.998
X2	1.000				0.972	0.554
X3	1.000				0.972	0.681
X4	1.000				0.972	0.789
sX =~						
X1	0.000				0.000	0.000



X2	1.000	0.100	0.057
X3	2.000	0.200	0.140
X4	3.000	0.300	0.243
iY =~			
Y1	1.000	1.032	0.987
Y2	1.000	1.032	0.409
Y3	1.000	1.032	0.545
Y4	1.000	1.032	0.620
sY =~			
Y1	0.000	0.000	0.000
Y2	1.000	0.092	0.037
Y3	2.000	0.184	0.097
Y4	3.000	0.276	0.166

Covariances:

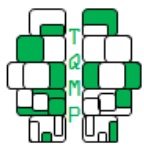
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
iX ~~						
sX	0.010	0.012	0.790	0.429	0.100	0.100
iY	0.007	0.042	0.170	0.865	0.007	0.007
sY	-0.001	0.015	-0.093	0.926	-0.015	-0.015
sX ~~						
iY	-0.007	0.010	-0.759	0.448	-0.070	-0.070
sY	-0.002	0.003	-0.475	0.635	-0.173	-0.173
iY ~~						
sY	0.032	0.042	0.762	0.446	0.334	0.334

Intercepts:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.X1	0.000				0.000	0.000
.X2	0.000				0.000	0.000
.X3	0.000				0.000	0.000
.X4	0.000				0.000	0.000
.Y1	0.000				0.000	0.000
.Y2	0.000				0.000	0.000
.Y3	0.000				0.000	0.000
.Y4	0.000				0.000	0.000
iX	3.964	0.040	99.655	0.000	4.078	4.078
sX	-0.802	0.009	-88.147	0.000	-8.020	-8.020
iY	0.054	0.043	1.276	0.202	0.053	0.053
sY	0.400	0.015	26.598	0.000	4.341	4.341

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
sX	0.010				1.000	1.000
.X1	0.004	0.022	0.198	0.843	0.004	0.005
.X2	2.109	0.124	17.024	0.000	2.109	0.684
.X3	1.013	0.063	16.120	0.000	1.013	0.497
.X4	0.425	0.035	12.069	0.000	0.425	0.280
.Y1	0.029	0.101	0.283	0.777	0.029	0.026
.Y2	5.211	0.312	16.680	0.000	5.211	0.821



.Y3	2.365	0.148	15.977	0.000	2.365	0.659
.Y4	1.437	0.158	9.100	0.000	1.437	0.519
iX	0.945	0.059	15.967	0.000	1.000	1.000
iY	1.064	0.118	8.991	0.000	1.000	1.000
sY	0.008	0.023	0.367	0.714	1.000	1.000

R-Square:

	Estimate
X1	0.995
X2	0.316
X3	0.503
X4	0.720
Y1	0.974
Y2	0.179
Y3	0.341
Y4	0.481

Annexe 12 : L'analyse de groupes multiples (*Multiple group analysis*)

Listing 12.1. Syntaxe pour réaliser une analyse de groupes multiples pour un modèle nul (étape 0 : modèle sans contrainte). Le fichier de données utilisé est `multiple_group_data_1.csv`, les variables utilisées sont VI1 à VI5 et GR puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 20 – Analyse de groupes multiples – Input .R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("multiple_group_data.csv", header = TRUE, sep = ",")

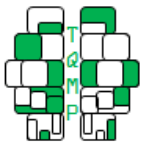
# Inspecter la base de données
str(data)

# Définir le modèle
# Modèle 1 - Modèle nul (sans contrainte)
model <- 'V2 ~ V1
          V4 ~ V3
          V5 ~ V4
          V5 ~ V2'

# Spécifier les paramètres du modèle
fit.1 <- sem(model, data = data, group = "GR")

# Obtenir le sommaire des estimations du modèle
summary(fit.1, fit.measures = TRUE)
```

Output 12.1. Sortie de l'analyse de groupes multiples du Listing 12.1. Le chiffre en **bleu** représente les degrés de libertés, les nombres en **jaune** sont les résultats du test du chi carré, puis les nombres en **vert** sont les indices d'ajustement.



```
> # Inspecter la base de données
> str(data)

' data.frame': 150 obs. of 6 variables:
 $ V1: num  0.291 0.735 0.409 0.16 0.432 -0.638 0.877 0.291 -0.466 -0.916 ...
 $ V2: num  3.965 -6.593 -5.184 0.959 1.222 ...
 $ V3: num  -1.163 -0.902 1.073 0.162 0.66 ...
 $ V4: num  17.2 6.55 -8.73 8.34 18.53 ...
 $ V5: num  -53.7 -41.3 25.5 -17.3 -44.8 ...
 $ GR: int   1 1 1 1 1 1 1 1 1 1 ...

> # Obtenir le sommaire des estimations du modèle
> summary(fit.1, fit.measures = TRUE)

lavaan 0.6-7 ended normally after 102 iterations
  Estimator                      ML
  Optimization method            NLMINB
  Number of free parameters      20
  Number of observations per group:
    1                      75
    0                      75

Model Test User Model:
  Test statistic                 3.793
  Degrees of freedom             10
  P-value (Chi-square)          0.956
  Test statistic for each group:
    1                      0.888
    0                      2.905

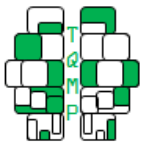
User Model versus Baseline Model:
  Comparative Fit Index (CFI)    1.000
  Tucker-Lewis Index (TLI)      1.030

Loglikelihood and Information Criteria:
  Loglikelihood user model (H0)  -1578.853
  Loglikelihood unrestricted model (H1) -1576.956

  Akaike (AIC)                  3197.706
  Bayesian (BIC)                 3257.919
  Sample-size adjusted Bayesian (BIC) 3194.623

Root Mean Square Error of Approximation:
  RMSEA                        0.000
  90 Percent confidence interval - lower 0.000
  90 Percent confidence interval - upper 0.000
  P-value RMSEA <= 0.05          0.977

Standardized Root Mean Square Residual:
  SRMR                        0.029 ...[]
```



Listing 12.2. Syntaxe pour réaliser une analyse de groupes multiples pour un modèle avec contraintes (étape 1 : coefficients de régression). Le fichier de données utilisé est `multiple_group_data_1.csv`, les variables utilisées sont VI1 à VI5 et GR puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 20 – Analyse de groupes multiples – Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("multiple_group_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Définir le modèle
model <- 'V2 ~ V1
          V4 ~ V3
          V5 ~ V4
          V5 ~ V2'

# Spécifier les paramètres du modèle
fit.2 <- sem(model, data = data, group = "GR", group.equal = c("regressions"))
```

```
# Obtenir le sommaire des estimations du modèle
summary(fit.2, fit.measures = TRUE)
```

Output 12.2. Sortie de l'analyse de groupes multiples du Listing 12.2. Les nombres en **jaune** sont les résultats du test du chi carré.

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.2, fit.measures = TRUE)
```

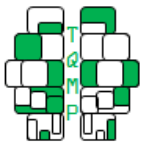
lavaan 0.6-7 ended normally after 87 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	20
Number of equality constraints	4
Number of observations per group:	
1	75
0	75

Model Test User Model:

Test statistic	13.178
Degrees of freedom	14
P-value (Chi-square)	0.513
Test statistic for each group:	
1	5.040
0	8.138

Model Test Baseline Model:



Test statistic	394.721
Degrees of freedom	18
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	1.000
Tucker-Lewis Index (TLI)	1.003

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-1583.545
Loglikelihood unrestricted model (H1)	-1576.956

Akaike (AIC)	3199.090
Bayesian (BIC)	3247.260
Sample-size adjusted Bayesian (BIC)	3196.623

Root Mean Square Error of Approximation:

RMSEA	0.000
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.106
P-value RMSEA <= 0.05	0.682

Standardized Root Mean Square Residual:

SRMR	0.073 ...[]
------	-------------

Listing 12.3. Syntaxe pour réaliser une analyse de groupes multiples pour un modèle avec contraintes (étape 2 : ordonnées à l'origine). Le fichier de données utilisé est `multiple_group_data_1.csv`, les variables utilisées sont VI1 à VI5 et GR puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 20 - Analyse de groupes multiples - Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

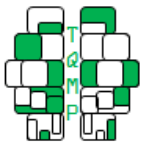
# Importer la base de données
data <- read.csv("multiple_group_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Définir le modèle
model <- 'V2 ~ V1
         V4 ~ V3
         V5 ~ V4
         V5 ~ V2'

# Spécifier les paramètres du modèle
fit.3 <- sem(model, data = data, group = "GR", group.equal = c("regressions", "
intercepts"))

# Obtenir le sommaire des estimations du modèle
```



```
summary(fit.3, fit.measures = TRUE)
```

Output 12.3. Sortie de l'analyse de groupes multiples du Listing 12.3. Les nombres en **jaune** sont les résultats du test du chi carré.

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.3, fit.measures = TRUE)

lavaan 0.6-7 ended normally after 60 iterations
  Estimator                      ML
  Optimization method            NLMINB
  Number of free parameters      20
  Number of equality constraints   7
  Number of observations per group:
    1                      75
    0                      75

Model Test User Model:
  Test statistic                15.896
  Degrees of freedom            17
  P-value (Chi-square)         0.531
  Test statistic for each group:
    1                      6.057
    0                      9.839

Model Test Baseline Model:
  Test statistic                394.721
  Degrees of freedom            18
  P-value                      0.000

User Model versus Baseline Model:
  Comparative Fit Index (CFI)    1.000
  Tucker-Lewis Index (TLI)      1.003

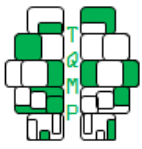
Loglikelihood and Information Criteria:
  Loglikelihood user model (H0)  -1584.904
  Loglikelihood unrestricted model (H1) -1576.956

  Akaike (AIC)                  3195.809
  Bayesian (BIC)                 3234.947
  Sample-size adjusted Bayesian (BIC) 3193.804

Root Mean Square Error of Approximation:
  RMSEA                        0.000
  90 Percent confidence interval - lower 0.000
  90 Percent confidence interval - upper 0.098
  P-value RMSEA <= 0.05          0.713

Standardized Root Mean Square Residual:
  SRMR                        0.079 ...[]
```

Listing 12.4. Syntaxe pour réaliser une analyse de groupes multiples pour un modèle avec contraintes (étape 3 : covariances résiduelles). Le fichier de données utilisé est `multiple_group_data_1.csv`, les variables utilisées sont VI1 à VI5 et GR puis la librairie nécessaire est `lavaan`. Le fichier de syntaxe R Chapitre 20 – Analyse de groupes



multiples - Input.R se trouve dans le matériel accompagnateur.

```
# Définir le répertoire de travail
setwd("chemin du répertoire dans lequel la base de données a été sauvegardée")

# Sélectionner le(s) librairie(s) nécessaire(s) pour l'analyse
library(lavaan)

# Importer la base de données
data <- read.csv("multiple_group_data.csv", header = TRUE, sep = ",")

# Inspecter la base de données
str(data)

# Définir le modèle
model <- 'V2 ~ V1
          V4 ~ V3
          V5 ~ V4
          V5 ~ V2'

# Spécifier les paramètres du modèle
fit.4 <- sem(model, data = data, group = "GR", group.equal = c("regressions", "
intercepts", "residual.covariances"))

# Obtenir le sommaire des estimations du modèle
summary(fit.4, fit.measures = TRUE)
```

Output 12.4. Sortie de l'analyse de groupes multiples du Listing 12.4. Les nombres en **jaune** sont les résultats du test du chi carré.

```
> # Obtenir le sommaire des estimations du modèle
> summary(fit.4, fit.measures = TRUE)
```

lavaan 0.6-7 ended normally after 60 iterations

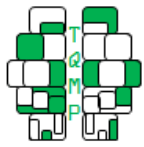
Estimator	ML
Optimization method	NLMINB
Number of free parameters	20
Number of equality constraints	7
Number of observations per group:	
1	75
0	75

Model Test User Model:

Test statistic	15.896
Degrees of freedom	17
P-value (Chi-square)	0.531
Test statistic for each group:	
1	6.057
0	9.839

Model Test Baseline Model:

Test statistic	394.721
Degrees of freedom	18



P-value	0.000
User Model versus Baseline Model:	
Comparative Fit Index (CFI)	1.000
Tucker-Lewis Index (TLI)	1.003
Loglikelihood and Information Criteria:	
Loglikelihood user model (H0)	-1584.904
Loglikelihood unrestricted model (H1)	-1576.956
Akaike (AIC)	3195.809
Bayesian (BIC)	3234.947
Sample-size adjusted Bayesian (BIC)	3193.804
Root Mean Square Error of Approximation:	
RMSEA	0.000
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.098
P-value RMSEA <= 0.05	0.713
Standardized Root Mean Square Residual:	
SRMR	0.079 [...]

Open practices

📄 The *Open Material* badge was earned because supplementary material(s) are available on [the journal's web site](#).

Citation

DION, K., BODNARUC, A. M., TRUDEL, G., LAMARCHE, J., RANGER, V., FOBERT, S., ... RENÉ, J.-L. (2021). La modélisation par équations structurelles – Un guide d'accompagnement pour l'interface R. *The Quantitative Methods for Psychology*, 17(3), 198-271. doi :[10.20982/tqmp.17.3.p198](https://doi.org/10.20982/tqmp.17.3.p198)

Copyright © 2021, *DION et collègues*. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Received: 10/01/2020 ~ Accepted: 26/06/2021