

Avant-propos

Objectifs

Ce manuel décrit toutes les étapes de l'expérimentation en psychologie, depuis la manipulation des facteurs, jusqu'à l'analyse statistique et la modélisation des données, en passant par la création et la présentation automatisée de stimuli. L'objectif principal de ce manuel est de montrer l'intérêt des logiciels pour automatiser diverses étapes de l'expérimentation pour lesquelles des opérations peuvent rapidement s'avérer répétitives. Nous insistons sur les nombreuses vertus de MATLAB[®] afin de montrer comment des algorithmes simples permettent de compiler des fichiers de données (au lieu d'ouvrir les fichiers les uns après les autres pour procéder à des copier-coller laborieux), générer des stimuli (au lieu de les dessiner un par un dans un logiciel de dessin) et recoder ou générer des tableaux de données.

Une seconde partie concerne la modélisation en psychologie, qui consiste à déterminer si un modèle est susceptible de s'ajuster aux données, en faisant varier ses paramètres par des simulations. L'objectif global est de montrer que l'algorithmique est essentielle à toutes les étapes de la production de données scientifiques.

Les auteurs ont fourni les fichiers MATLAB[®] pertinents pouvant être utilisés en parallèle de l'ouvrage¹.

Atouts de la programmation

Utiliser des logiciels standards en psychologie tels qu'E-Prime et SPSS, pour expérimenter puis analyser les données, peut rapidement présenter une limite. Si vous visez une expérimentation simple et une analyse simple élémentaire des données, les

1. Ils sont téléchargeables au lien suivant : <https://doi.org/10.5281/zenodo.1339342>.

logiciels habituellement utilisés en psychologie (par exemple, E-Prime, Excel, SPSS, Statistica) ont une utilité indéniable et peuvent être suffisants pour remplir vos objectifs. MATLAB[®] peut néanmoins s'avérer un outil très simple d'utilisation pour expérimenter et analyser vos données, si le temps requis pour apprendre son langage ne constitue pas un frein. Il peut donc apparaître que son apprentissage n'est pas nécessaire à court terme lorsque les logiciels précédemment cités semblent suffire. Néanmoins, notre conseil est de faire le choix à long terme de la programmation, afin d'accélérer toutes les étapes de la production de données et augmenter la fiabilité de vos données. La connaissance d'un langage de programmation tel que MATLAB[®] permet, en outre, de développer l'imagination et d'envisager des expérimentations et des analyses plus raffinées. MATLAB[®] est une façon de décupler votre créativité dans le domaine de la recherche.

Sur le plan de l'efficacité, MATLAB[®] est clairement un atout. Programmer une analyse de données avec MATLAB[®] présente l'intérêt de pouvoir refaire l'analyse en un seul clic si besoin. Après avoir spécifié une liste de commandes visant à enchaîner les différentes étapes de l'analyse descriptive, les étapes de la transformation des données et les étapes de l'analyse inférentielle, on dispose d'un programme réutilisable à l'envie. Imaginons qu'un dossier contienne 50 fichiers correspondant aux données de 50 participants. Un programme peut automatiquement lister les fichiers, créer une variable $N = 50$, faire un histogramme des performances (les temps de réponse, par exemple), transformer les temps de réponse par un logarithme pour éviter leur distribution asymétrique et augmenter la puissance des tests inférentiels qui suivront, etc. Toutes ces étapes de programmation mises bout à bout peuvent correspondre à autant d'étapes avec un logiciel de type SPSS. Cependant, en cas de modification des données, MATLAB[®] présente l'énorme avantage qu'il suffit de relancer le programme d'un seul clic de souris (puisque'il est déjà fait) pour reproduire la même séquence d'analyses, alors que SPSS requiert de refaire chacune des étapes de l'analyse².

Repasser par toutes les étapes de l'analyse en cliquant à nouveau dans les menus peut être laborieux et chronophage. Pourtant, reconduire une analyse de données est assez fréquent, pour des raisons diverses : 1) on réalise après coup qu'un fichier présente des données aberrantes car le participant semble ne pas avoir compris la consigne et montre des performances non représentatives de l'échantillon, si bien que le participant doit être retiré de l'analyse ; 2) l'analyse manque de puissance et l'on souhaite augmenter la taille de l'échantillon (même si modifier la taille d'un échantillon jusqu'à obtenir une valeur significative n'est pas recommandé par les experts en statistique, c'est une méthode courante) ou 3) on souhaite conduire la

2. À moins de connaître le langage SYNTAX, permettant également de lister une suite de commandes, mais avec beaucoup moins de souplesse que MATLAB.

même analyse avec un échantillon différent (par exemple, des enfants à la place d'adultes) quelques mois plus tard. Quel que soit le cas, il suffira de mettre à jour le dossier de données afin que MATLAB[®] récupère les nouvelles données, qu'il mette à jour automatiquement la variable N pour appliquer l'ancien programme aux nouveaux fichiers et qu'il conduise l'analyse à terme, en produisant tout au long du processus les tableaux, les graphes et les analyses statistiques. La planification de l'analyse, l'enchaînement logique des analyses et l'ensemble des paramètres esthétiques des tableaux et des figures sont conservés. Au contraire, SPSS requiert une nouvelle planification de l'ensemble des analyses (on ne se souvient plus nécessairement de l'ordre dans lequel on a cliqué dans les divers menus) et cela oblige à double-cliquer sur chacune des nouvelles figures pour les éditer (par exemple, transformer des barres colorées en barres hachurées, changer l'échelle des axes pour améliorer les proportions, rajouter les barres d'erreurs, etc.). Programmer sous MATLAB[®] peut prendre une journée, ce qui n'est pas nécessairement plus rapide qu'avec SPSS. L'avantage accordé ici à l'utilisation de MATLAB[®] peut donc se comprendre aisément par le fait que la moindre modification des données requiert une journée de plus avec SPSS, mais seulement quelques secondes de plus avec MATLAB[®].

Un second exemple est lié à l'application de règles pour recoder des données. On souhaite par exemple créer, après coup, une nouvelle variable afin de recoder une condition expérimentale. Cette variable prendrait par exemple la valeur 1 (signifiant que la condition est réussie) si le participant a appuyé sur la touche x à l'essai n , après avoir appuyé sur la touche y à l'essai $n-1$, uniquement lorsque le stimulus z a été affiché dans la partie nord de l'écran et uniquement lorsque le temps de présentation était inférieur à 100 ms. Cette opération est difficile sans une recherche par un algorithme permettant de comparer les données sur des lignes et des colonnes différentes (SPSS ne recode des données qu'en procédant à des tests sur une même ligne ; par exemple, si genre = homme, alors coder 1).

Enfin, par exemple, lorsque l'on souhaite tester un modèle afin de trouver les paramètres les mieux ajustés aux données, il est parfois nécessaire de lire un tableau de données des milliers de fois pour calculer la meilleure adéquation du modèle aux données. Cette méthode n'est pas possible avec un logiciel de statistiques classique dans lequel sont répertoriées le plus souvent des procédures classiques (le plus souvent, de simples tests statistiques et non des modèles visant à expliquer des processus mentaux tels que des modèles de la mémoire ou des processus de décision). MATLAB[®] vous permettra de développer ou tester des modèles originaux spécifiques à votre domaine d'étude.

Commencer par les chapitres sur la modélisation et l'expérimentation est conseillé à ceux souhaitant repousser le plus loin possible l'apprentissage de MATLAB[®], tout

en souhaitant se familiariser avec ces méthodes. Le chapitre initiation à la programmation permet de commencer l'apprentissage de MATLAB[®] et d'assimiler l'entièreté du contenu de tous les chapitres. Les chapitres demeurent néanmoins relativement indépendants les uns les autres et peuvent être parcourus globalement sans les rudiments de MATLAB[®].

Remerciements

Nous tenons à remercier particulièrement Emmanuelle Ménétrier pour sa relecture attentive, André Didierjean, Alessandro Guida, Lucie Laurent, Noelia Do Carmo Blanco, les étudiants de Nice ayant suivi le parcours Sciences Cognitives et le cours d'introduction aux sciences cognitives. Merci également aux membres de l'équipe Langage et Cognition et plus généralement du laboratoire Bases, Corpus, Langage (CNRS UMR 7320), pour le soutien et les commentaires apportés.