

Devoir Statistiques Avancées

Thomas Husson, Groupe 52

Table des matières

1 Énoncé et présentation des échelles	3
1.A Énoncé du devoir	3
1.B Échelles	3
2 Gestion des données	4
2.A Présentation des fichiers de données	4
2.B Import des données et data management	4
3 Question 1 : Validation de l'échelle Hamilton	6
3.A Validation à J0	6
3.B Validation à J56	15
4 Question 2 : Comparaison de la réponse au traitement entre deux groupes de patients	22
4.A Gestion des données	22
4.B Scores bruts à J0 dans chaque groupe	22
4.C Évolution des scores bruts entre J0 et J56 dans chaque groupe	23
4.D Comparaison de la réponse au traitement entre deux groupes de patients	25
4.E Conclusion sur la comparaison de la réponse au traitement entre les deux groupes de patients	31
5 Question 3 : Analyse de survie	32
5.A Critère binaire de réponse (50% à la dernière visite observée)	32
5.B Intégration du critère binaire de réponse dans une analyse de survie	32
5.C Analyse de survie avec le critère binaire de réponse	32
6 Conclusion générale	34
7 Annexe – Code R	36

! Important

Utilisation de l'IA

Des LLMs ont été utilisés à plusieurs reprises dans ce devoir, pour deux tâches principales :

- En cas de problème d'exécution du code R (pour suggérer correction et amélioration)
- Pour amélioration du rendu depuis un fichier Quarto Markdown vers PDF. Notamment certaines fonctions dont l'output R n'est pas compatible avec le rendu pdf (par exemple, `factanal()` pour l'analyse factorielle ou le rendu des tableaux automatisé des coefficients alpha de Cronbach et leur IC)

Utilisation de Gemini pour le rendu Quarto et Github Copilot pour le code R.

1 Énoncé et présentation des échelles

1.A Énoncé du devoir

Consigne :

- Étude d'épidémiologie clinique avec mesures répétées
- Données :
 - 146 patients déprimés
 - Évaluations à J0, J4, J7, J14, J21, J28, J42, J56
 - Autoévaluation (SCL90) et hétéroévaluation (échelle de dépression de Hamilton)
- Questions :
 1. Validation de l'échelle de dépression de Hamilton aux temps J0 et J56
 2. Comparaison de la réponse au traitement entre deux groupes de patients (groupe=0 et groupe=1) en utilisant le score brut de Hamilton avec une approche LOCF puis un modèle mixte
 3. Réponse à la question 2 en utilisant un critère binaire censuré « réponse au traitement » défini par une chute de 50% à l'échelle de Hamilton par rapport à J0
- Fichiers :
 - Fichier groupe (`outil groupe.xlsx`) (2 sous-groupes de patients)
 - Fichier autoévaluation (`outil autoeval.xlsx`) (SCL 90)
 - Fichier hdrs (`outil hdrs.xlsx`) (échelle de Hamilton)

1.B Échelles

Table 1 : Présentation des échelles utilisées dans le devoir

	Échelle de Hamilton (HDRS)	Échelle SCL90
Objectif	Mesure l'intensité de la symptomatologie dépressive	"Inconfort psychopathologique" selon plusieurs dimensions.
Type	Hétéro-évaluation	Autoévaluation
Méthode	17 items codés de 2 à 4- Score ≤ 7 : pas de dépression clinique- Score 8-15 : dépression mineure- Score > 15 : dépression majeure	10 dimensions : somatisation, symptômes obsessionnels, sensibilité interpersonnelle, dépression, anxiété, hostilité, phobies, traits paranoïaques, traits psychotiques et symptômes divers.

2 Gestion des données

2.A Présentation des fichiers de données

Les 3 fichiers sont en format “large” : chaque ligne correspond à une visite d’un patient et une colonne par item de l’échelle (sauf l’item 16 = PERTE DE POIDS qui est codé en deux variables HAMD16A et HAMD16B dans l’échelle de Hamilton, selon que la perte de poids est déclarée par le patient ou appréciée par le médecin)

On créera donc une colonne `hdrs$HAMD16` qui prendra la valeur de `hdrs$HAMD16A` si elle est remplie, sinon la valeur de `hdrs$HAMD16B`.

2.A.1 Fichier Hamilton

- 1053 observations, 20 variables pour 146 patients
- On ajoute une colonne `score` qui contient le score total de l’échelle de Hamilton (somme des items)
- Les données d’une ligne (J7 du 128ème patient) sont manquantes → on supprime cette ligne.

2.A.2 Fichier scl90

- 1034 observations, 92 variables, 146 patients.
- On crée 10 nouvelles variables représentant les scores moyen des 10 dimensions de l’échelle SCL90.
- Données aberrantes parfois, qui sont recodées en données manquantes et représentent ainsi 0.6% des données totales.

2.A.3 Fichier groupe

- Répartit les 146 patients en 2 groupes (1 et 0)
- Pas de NA

2.B Import des données et data management

Les données sont importées à partir de fichiers Excel.

2.B.1 SCL90

Le jeu de données `scl90` est traité de la manière suivante :

- Visites ordonnées chronologiquement
- Identification des doublons
- Visualisation et gestion des données aberrantes
- Imputation des données manquantes par le mode pour chaque question
- Création des scores moyens par dimension (10 dimensions)
- Nouveau dataframe `scl90_dim` avec uniquement les 10 dimensions

2.B.2 HDRS

Le jeu de données `hdrs` est traité de la manière suivante :

- Visites ordonnées chronologiquement
- Identification des doublons
- Fusion des variables HAMD16A et HAMD16B en une seule variable HAMD16
- Création du score total HDRS (ajouté dans la colonne `hdrs$score`)

2.B.3 Groupes

2.B.4 Fusion des 3 fichiers

Convertir `hdrs_groupe`, `scl90_groupe` et `df_total_wide` de format “large” à format “long”

3 Question 1 : Validation de l'échelle Hamilton

Note

Consigne de la question 1 : Lorsque l'on utilise un instrument de mesure subjective dans une étude clinique, il est toujours bon de le (re)valider rapidement. Procédez ici à cette **vérification** sur l'échelle de dépression de Hamilton, aux temps J0 et J56.

- Vérification d'une échelle de mesure subjective = 1/ Que mesure l'instrument ? 2/ Que vaut la mesure ?
- Premier temps : Évaluation préliminaire des réponses aux items, puis chercher une corrélation entre eux par une matrice de corrélation 2 à 2
- Second temps : Analyse de la structure dimensionnelle = **que mesure l'instrument ?**
 - Exploration de la structure par analyse en composante principale : visualiser les relations entre les items
 - Détermination du nombre de dimensions : diagramme des valeurs propres (*scree plot*) permet de déterminer le nombre de dimensions (composantes principales)
 - Si structure dimensionnelle identifiée : **analyse factorielle** permet de déterminer quels items se regroupent dans chaque dimension
- Troisième temps : Évaluation de la fiabilité interne = **que vaut la mesure ?**
 - La consistance interne des items (évalue si les items sont cohérents entre eux) sera évaluée par le calcul de l'alpha de Cronbach, calculé sur l'échelle totale et sur chaque dimension identifiée précédemment.
- Quatrième temps : Évaluation de la validité = **l'instrument mesure-t-il ce qu'il est censé mesurer ?** (similaire à la question "que mesure l'instrument ?")
 - Validité interne : déjà évaluée au cours du second temps (structure dimensionnelle)
 - Validité externe : corrélation avec d'autres instruments de mesure (ici les dimensions de l'échelle SCL90)

3.A Validation à J0

3.A.1 Description

Les réponses sont représentées :

- par des diagrammes en bâtons pour chaque item de l'échelle de Hamilton à J0
- par une matrice de corrélation 2 à 2 entre les items

NB : le code R utilise une fonction pour faciliter la création des diagrammes en bâtons pour chaque item.

La fonction crée un diagramme en bâtons pour chaque item listés dans un vecteur crée précédemment (`hdrs_items`).

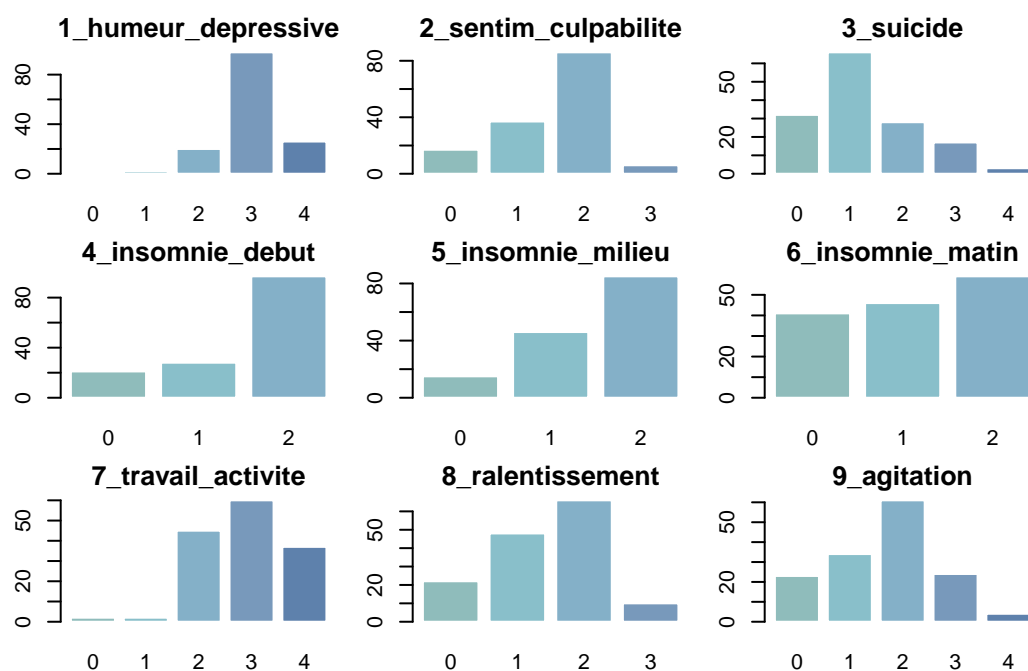


FIGURE 1 – Diagrammes en bâtons des scores des items de l'échelle de Hamilton à J0

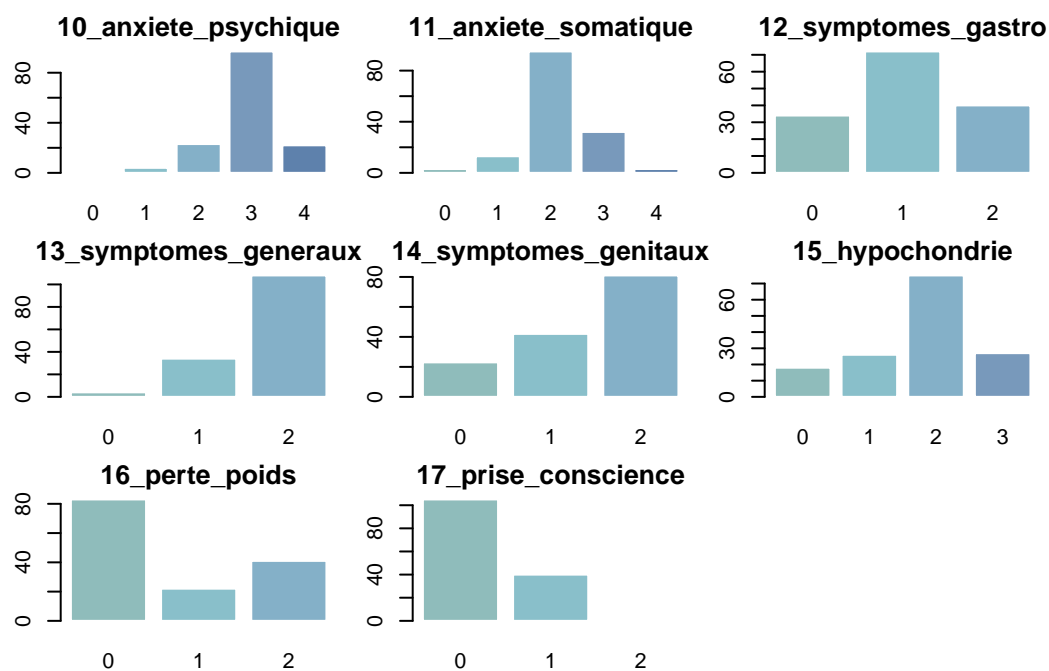


FIGURE 2 – Diagrammes en bâtons des scores des items de l'échelle de Hamilton à J0

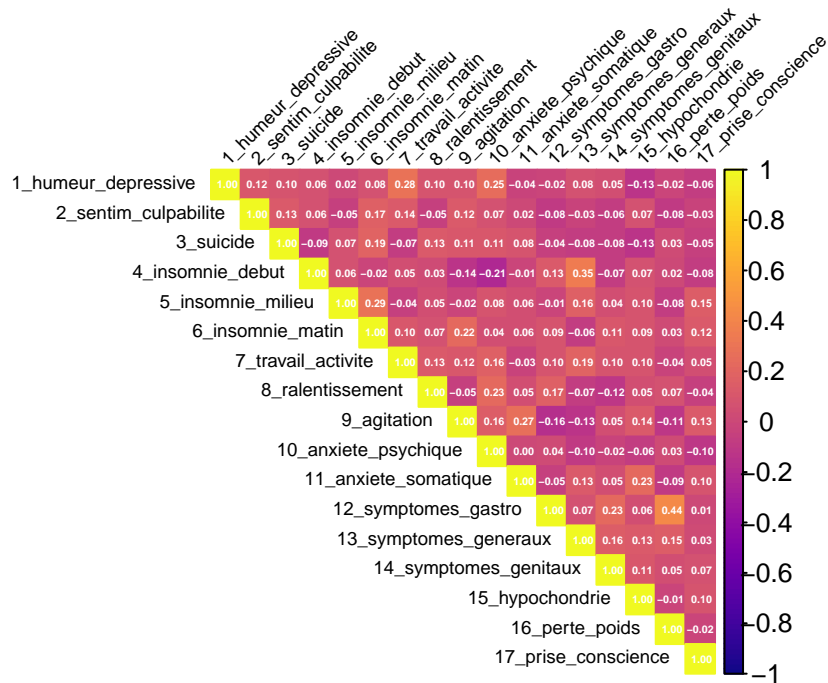


FIGURE 3 – Matrice de corrélation entre les items de l'échelle de Hamilton à J0

- Il n'y a pas de données manquantes.
- Les diagrammes en bâtons montrent que certains items ont une distribution asymétrique (ex : insomnie quel que soit le moment de la nuit, symptômes généraux, perte de poids...)
- On observe un effet plancher marqué pour certains items (notamment perte de poids et prise de conscience), avec une majorité de scores à 0.
- La matrice de corrélation des items 2 à 2 ne retrouve pas de coefficient de corrélation supérieure à 0,50 en valeur absolue, il n'existe pas de redondance entre les items de l'échelle Hamilton.

3.A.2 Validité interne : structure dimensionnelle, analyse factorielle

3.A.2.1 Exploration de la structure dimensionnelle : analyse en composantes principales

- On peut réaliser une analyse en composantes principales (ACP) pour visualiser les relations entre les items.

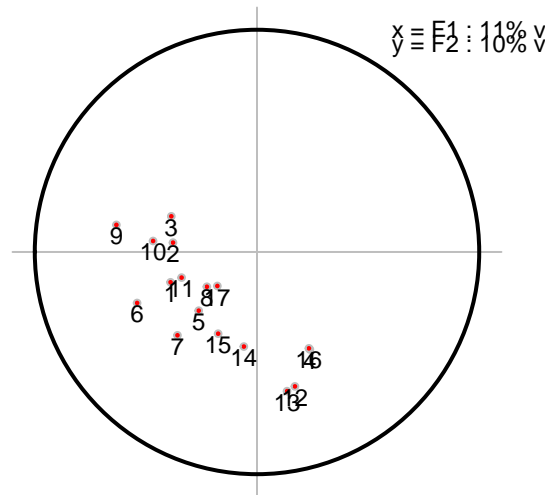


FIGURE 4 – Analyse en composantes principales des items de l'échelle de Hamilton à J0

- Chaque point représente un item de l'échelle de Hamilton.
- Deux axes principaux :
 - l'axe horizontale x représente la première composante principale (PC1) qui explique 11% de la variance totale, l'axe verticale y représente la deuxième composante principale (PC2) qui explique 10% de la variance totale.
 - Ensemble, les deux premières composantes principales expliquent 21% de la variance totale, ce qui est relativement faible.
- La majorité des variables sont proches du centre, ce qui indique qu'elles ne contribuent pas fortement aux premières composantes principales.
- Au total, cette ACP ne révèle pas de structure dimensionnelle claire parmi les items de l'échelle de Hamilton à J0.

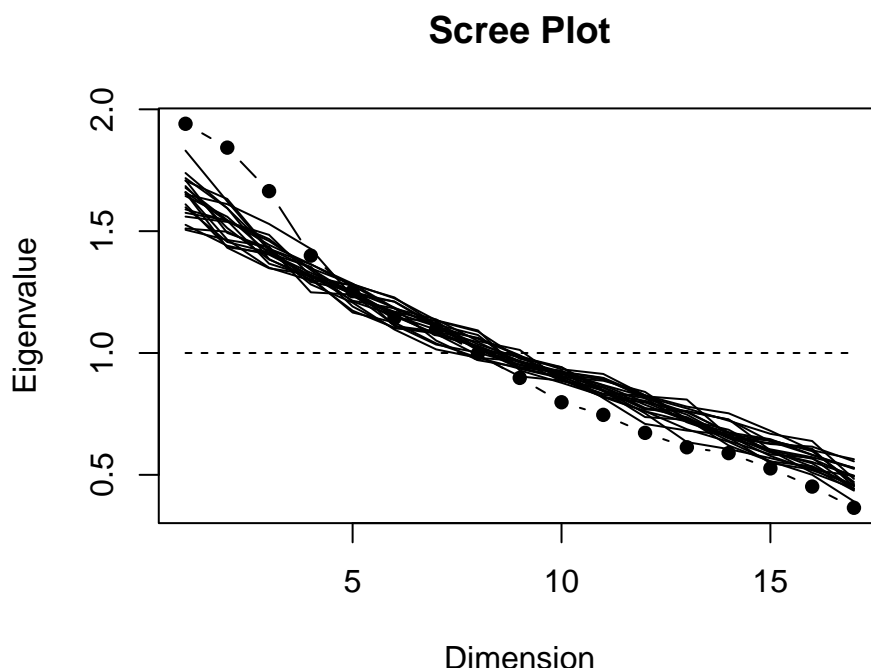


FIGURE 5 – Diagramme des valeurs propres (scree plot) des items de l’échelle de Hamilton à J0 avec représentation de données simulées (analyse parallèle)

3.A.2.2 Exploration de la structure dimensionnelle : analyse factorielle

- À J0, le *scree plot* ne permet pas d’identifier un nombre clair de facteurs : les valeurs propres décroissent progressivement sans “coude” net.
- En analyse parallèle, on observe au moins 3 dimensions ayant une valeur propre supérieure à celle obtenue sur des données simulées.
- On pourrait réaliser des tests statistiques qui permettraient de déterminer le nombre optimal de dimensions, mais ces tests sont sujets à plusieurs biais :
 - on calculerait une p-value pour l’hypothèse “n facteurs sont suffisants”
 - mais ces tests sont difficiles à interpréter et sensibles à la taille de l’échantillon
- On retient donc 3 facteurs principaux pour l’analyse factorielle.

TABLE 2 – Contribution des facteurs à la variance de la réponse à chaque item du score de Hamilton évalué à J0 (analyse factorielle avec rotation varimax à 3 facteurs)

Variable	Factor1	Factor2	Factor3
1_humeur_depressive	0.006	0.028	0.263
2_sentim_culpabilite	-0.057	-0.076	0.240
3_suicide	-0.021	-0.120	0.207
4_insomnie_debut	0.110	0.359	-0.050
5_insomnie_milieu	0.004	0.118	0.240
6_insomnie_matin	0.132	-0.155	0.427
7_travail_activite	0.124	0.121	0.320
8_ralentissement	0.183	-0.105	0.101
9_agitation	-0.107	-0.228	0.501
10_anxiete_psychique	0.066	-0.152	0.255

Variable	Factor1	Factor2	Factor3
11_anxiete_somatique	-0.029	0.070	0.316
12_symptomes_gastro	0.992	0.044	-0.091
13_symptomes_generaux	0.046	0.977	0.194
14_symptomes_genitaux	0.238	0.120	0.147
15_hypochondrie	0.076	0.089	0.226
16_perte_poids	0.424	0.155	-0.100
17_prise_conscience	0.022	-0.004	0.176

- À J0, l’analyse factorielle exploratoire avec rotation varimax met en évidence 3 facteurs latents expliquant cumulativement 21,9 % de la variance des réponses aux items du score de Hamilton.
- Concernant l’interprétation du tableau des “loadings” (charges factorielles) :
 - Les “loadings” (charges factorielles) indiquent la contribution de chaque item à chaque facteur. Ils mesure la force de la régression linéaire entre chaque item et chaque facteur latent. Plus l’item est proche de 1, plus il est expliqué par le facteur latent commun.
 - Il n’existe pas de “seuil” universel pour déterminer si un loading est “élevé” ou “faible”, mais généralement, un loading supérieur à 0,4 ou 0,5 est considéré comme significatif.
- On peut interpréter les 3 facteurs identifiés comme suit :
 - Facteur 1 : 8, 12, 14, 16 (principalement des symptômes somatiques).
 - Facteur 2 : 4, 13 (relatifs à l’asthénie).
 - Facteur 3 : les items restants (symptômes dépressifs psychiatriques proprement dits).

On peut rajouter 3 “sous-scores” au score total de Hamilton à J0, correspondant aux scores moyens des items chargés sur chaque facteur.

- A titre exploratoire, on peut refaire une ACP sur ces 3 sous-scores pour visualiser leur relation.

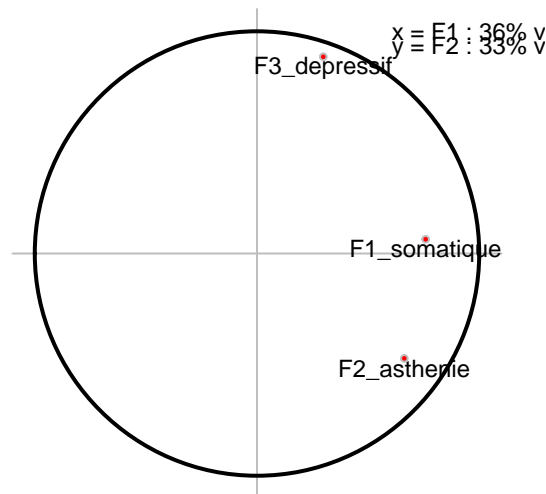


FIGURE 6 – Analyse en composantes principales des sous-scores de l’échelle de Hamilton à J0

- Les 3 sous-scores sont bien représentés (proches du cercle). Le facteur “symptômes dépressifs” semble orthogonal aux deux autres facteurs.
- La variance totale expliquée par ces 3 sous-score est de 69%.

Une ACP focalisée sur ces 3 sous-scores et le score total de Hamilton permet de visualiser la relation entre le score total et les sous-scores.

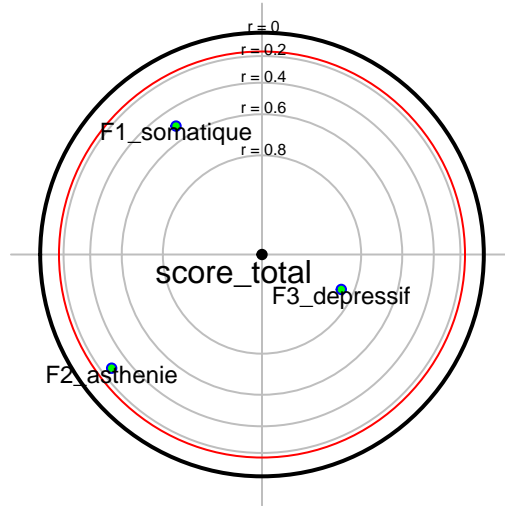


FIGURE 7 – Analyse en composantes principales focalisée des sous-scores de l'échelle de Hamilton à J0

Le score total de Hamilton semble plus corrélé aux symptômes dépressifs (F3) qu'aux deux autres sous-scores.

3.A.3 Fiabilité interne = que vaut la mesure ?

3.A.3.1 Consistance interne : alpha de Cronbach La consistance interne des items de l'échelle de Hamilton à J0 est évaluée par le calcul de l'alpha de Cronbach, qui correspond globalement au **pourcentage de « variance partagée »** entre le score vrai (hypothétique) et la mesure obtenue.

Il permet ainsi de mesurer la cohérence entre les items d'une échelle de mesure, et est élevé lorsque les items sont fortement corrélés entre eux.

On peut donc calculer dans un premier temps l'alpha de Cronbach sur l'ensemble des items de l'échelle de Hamilton à J0, puis sur chacun des 3 facteurs identifiés précédemment.

Les intervalles de confiance (IC) à 95% des alpha de Cronbach sont estimés par la méthode du bootstrap avec 1000 rééchantillonnages. Le bootstrap est possible ici car il y a > 100 observations.

TABLE 3 – Alpha de Cronbach et intervalles de confiance à 95% pour l'échelle de Hamilton à J0 et ses sous-échelles

Scale	Alpha	CI_lower	CI_upper
Global	0.456	0.262	0.624
F1_somatique	0.381	0.188	0.530
F2_asthenie	0.490	0.266	0.653
F3_depressif	0.486	0.349	0.622

Au total, quel que soit le niveau d'analyse (global ou par facteur), les alpha de Cronbach sont < 0.5 , indiquant une faible consistance interne des items de l'échelle de Hamilton à J0.

Ce résultat est cohérent avec une structure multidimensionnelle, l'alpha étant pénalisé lorsqu'une échelle ne mesure pas un seul trait latent.

3.A.4 Validité externe = l'instrument mesure-t-il ce qu'il est censé mesurer ?

- Validité externe d'un instrument cherche à démontrer que l'instrument se comporte logiquement par rapport au réseau théorique qui lui est associé.
- Selon la théorie nomologique (c'est à dire selon les relations postulées entre les différents concepts d'une même discipline), la dépression mesurée par l'échelle de Hamilton doit être fortement liée à d'autres manifestations de la détresse psychologique générale (mesurée par le SCL-90), mais distincte de certains autres concepts.
- Ici, la validité du construit peut être évaluée en évaluant la validité convergente (corrélation forte entre des concepts proches).
- Il est plus difficile d'évaluer la validité divergente (corrélation faible entre des concepts différents) car le SCL-90 mesure principalement des dimensions de la détresse psychologique ; de même pour la validité concurrente (corrélation forte avec un *gold-standard*, car nous ne disposons pas d'un instrument de mesure de la dépression reconnu comme un *gold-standard* ici).

3.A.4.1 Corrélation entre le score de Hamilton et les dimensions du SCL-90 à J0

- On peut représenter une matrice de corrélation entre le score total de Hamilton et les 10 dimensions du SCL-90 à J0.



FIGURE 8 – Corrélation entre le score total de l'échelle de Hamilton et les dimensions du SCL-90 à J0

La corrélation est au maximum à 0.37 avec la composante “somatisation” du SCL-90 et de 0.40 avec la composante “symptômes divers” du SCL-90, indiquant une validité convergente modérée entre le score total de Hamilton et cette dimension du SCL-90 à J0.

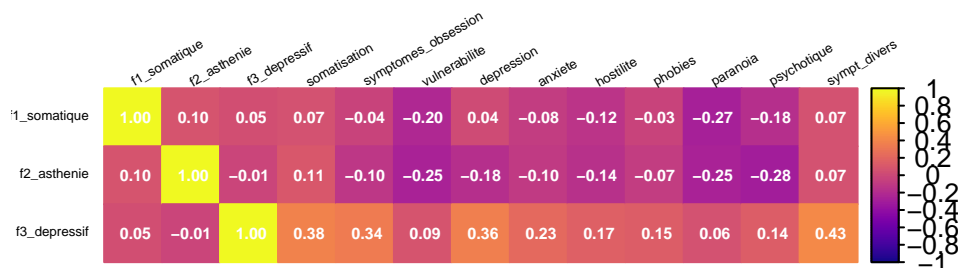


FIGURE 9 – Corrélation entre les sous-scores de l'échelle de Hamilton et les dimensions SCL-90 à J0

- Il n’y a pas non plus de corrélation forte entre les sous-scores de Hamilton et les dimensions du SCL-90 à J0.
- Par exemple, le sous-score “symptômes somatiques” de Hamilton est faiblement corrélé avec la dimension “somatisation” du SCL-90 ($r = 0.07!!$).

3.A.4.2 Conclusion validité externe à J0 La validité convergente entre le score total de Hamilton et les dimensions du SCL-90 à J0 est faible à modérée, suggérant que l’échelle de Hamilton mesure partiellement des aspects de la détresse psychologique générale, mais pas de manière très forte.

3.A.5 Conclusion globale à J0

À J0, l’échelle de dépression de Hamilton présente une structure dimensionnelle peu claire, avec une faible consistance interne des items (α de Cronbach < 0.5) et une validité convergente modérée avec les dimensions du SCL-90.

Ces résultats suggèrent que l’échelle de Hamilton pourrait ne pas être un instrument optimal pour mesurer la dépression dans cette population à ce moment précis.

3.B Validation à J56

3.B.1 Description

À J56, 120 des 146 patients présents à J0 ont répondu à tous les items : 17.8% des patients ont été perdus de vue.

Comme à J0, les réponses sont représentées :

- par des diagrammes en bâtons pour chaque item de l'échelle de Hamilton à J56
- par une matrice de corrélation 2 à 2 entre les items

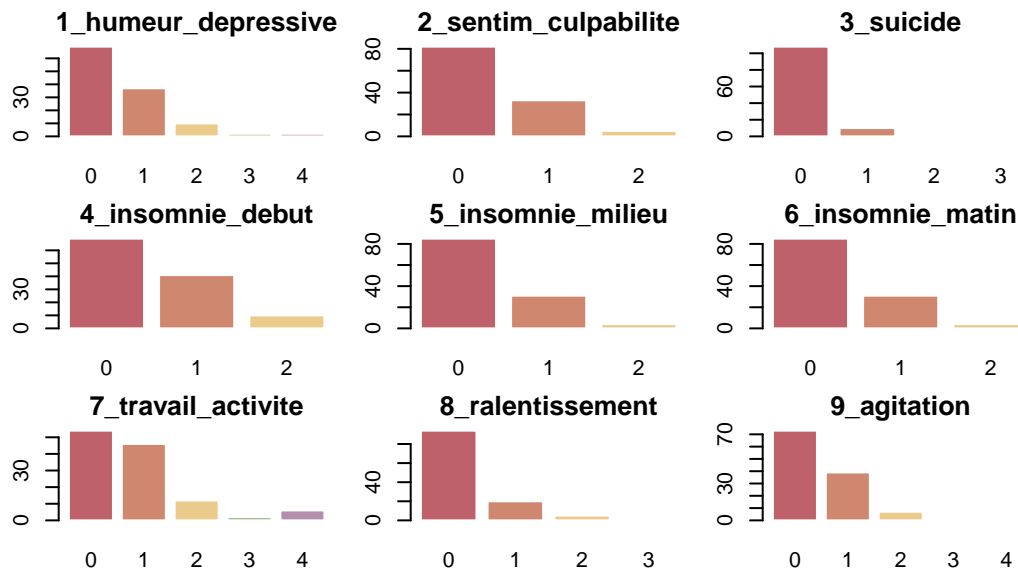


FIGURE 10 – Diagrammes en bâtons des scores des items de l'échelle de Hamilton à J56

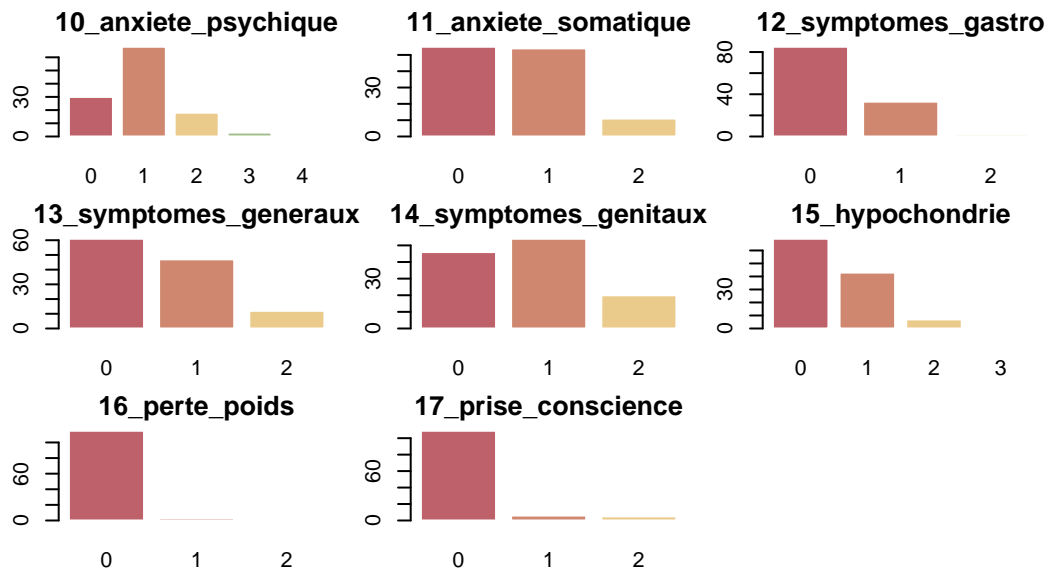


FIGURE 11 – Diagrammes en bâtons des scores des items de l'échelle de Hamilton à J56

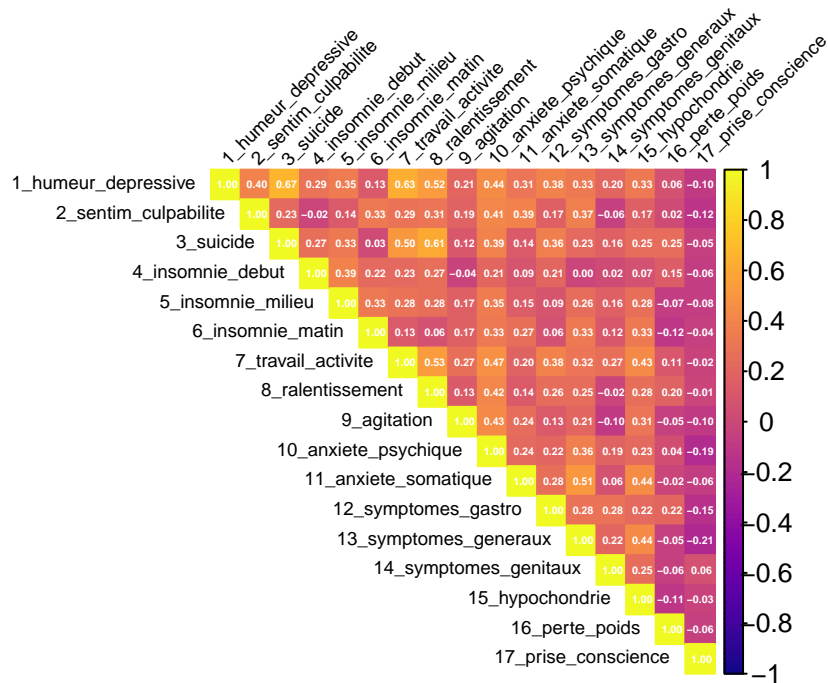


FIGURE 12 – Matrice de corrélation entre les items de l'échelle de Hamilton à J56

- Sur les diagrammes en bâtons, on observe un effet plancher, c'est à dire que de nombreux patients ont des scores faibles sur plusieurs items. Il est probable que ce soit lié à l'amélioration de la symptomatologie dépressive entre J0 et J56.
- Cet effet plancher peut augmenter mécaniquement certaines corrélations entre items, ce qui doit être gardé en tête pour l'interprétation.
- La matrice de corrélation des items met en évidence des coefficients de corrélation plus élevés que ceux observés à J0, avec plusieurs coefficients supérieurs à 0.5 en valeur absolue, indiquant une certaine redondance entre les items de l'échelle Hamilton à J56.
- Il est possible que cette redondance soit liée à l'effet plancher observé sur les diagrammes en bâtons, où de nombreux patients ont des scores faibles sur plusieurs items, ce qui peut augmenter artificiellement les corrélations entre ces items.

3.B.2 Validité interne : structure dimensionnelle, analyse factorielle

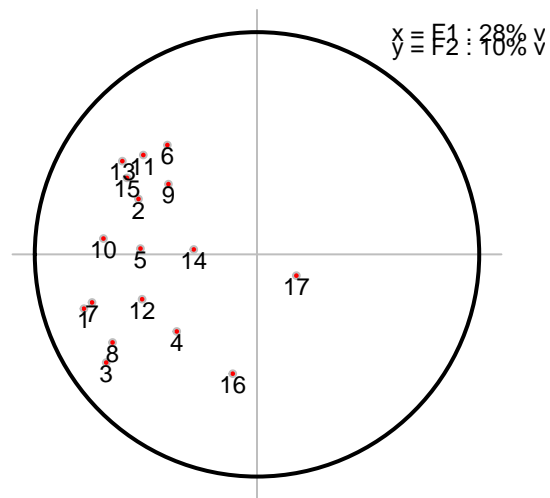


FIGURE 13 – Analyse en composantes principales des items de l'échelle de Hamilton à J56

3.B.2.1 Exploration de la structure dimensionnelle : analyse en composantes principales

- La première composante principale explique 28% de la variance totale, la deuxième composante principale explique 10% de la variance totale.
- Les deux premières composantes principales expliquent 38% de la variance totale, ce qui est une amélioration par rapport à J0.
- Certaines variables sont un peu plus éloignées du centre qu'à J0, il est possible qu'elles contribuent davantage aux premières composantes principales.

Scree Plot

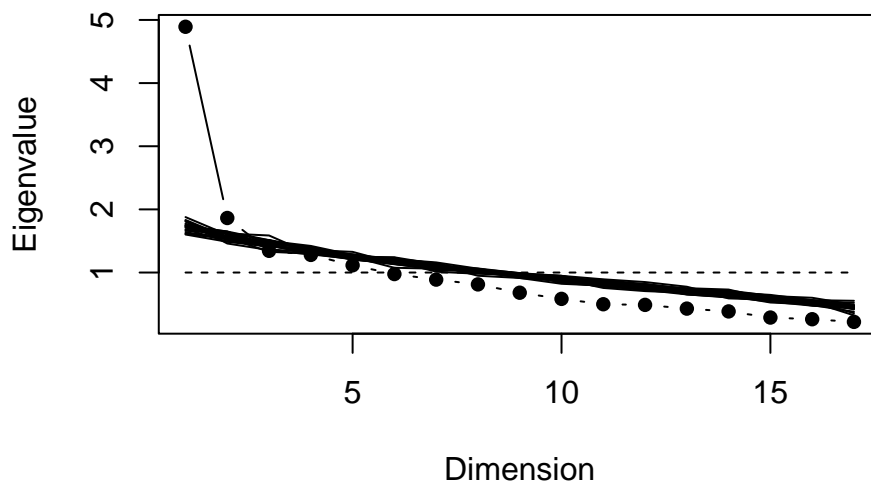


FIGURE 14 – Diagramme des valeurs propres (scree plot) des items de l'échelle de Hamilton à J56 avec représentation de données simulées (analyse parallèle)

3.B.2.2 Exploration de la structure dimensionnelle : analyse factorielle

- À J56, le *scree plot* n'est pas du tout le même qu'à J0 : on observe un "coude" net après la première valeur propre.
- **L'échelle Hamilton à J56 semble donc essentiellement unidimensionnelle.**

TABLE 4 – Contribution du facteur unique à la variance de la réponse à chaque item du score de Hamilton évalué à J56 (analyse factorielle avec rotation varimax à 1 facteur)

Variable	Factor1
1_humeur_depressive	0.802
2_sentim_culpabilite	0.468
3_suicide	0.706
4_insomnie_debut	0.333
5_insomnie_milieu	0.456
6_insomnie_matin	0.282
7_travail_activite	0.745
8_ralentissement	0.657
9_agitation	0.328
10_anxiete_psychique	0.624
11_anxiete_somatique	0.401
12_symptomes_gastro	0.471
13_symptomes_generaux	0.495
14_symptomes_genitaux	0.252
15_hypochondrie	0.494
16_perte_poids	0.125
17_prise_conscience	-0.132

- Les valeurs numériques mesurent la force de la relation linéaire entre chaque item observé et le facteur latent unique.
- Les symptômes centraux de la dépression (humeur dépressive, ralentissement, baisse d'activité) présentent les charges factorielles les plus élevées, indiquant qu'ils sont de bons indicateurs du facteur latent unique (la dépression).
- Certains items périphériques (perte de poids, symptômes génitaux) ont une très faible charge, ils peuvent être peu informatifs et faiblement contribuer à la mesure du concept unique de dépression à J56. Leur pertinence dans le score total de l'instrument peut être remise en question.

3.B.3 Fiabilité interne = que vaut la mesure ?

3.B.3.1 Consistance interne : alpha de Cronbach La consistance interne des items de l'échelle de Hamilton à J56 est évaluée par le calcul de l'alpha de Cronbach, et son IC à 95% par la méthode du bootstrap avec 1000 rééchantillonnages.

TABLE 5 – Alpha de Cronbach et intervalle de confiance à 95% pour l'échelle de Hamilton à J56

Scale	Alpha	CI_lower	CI_upper
Global	0.82	0.76	0.86

- A J56, l'alpha de Cronbach est de 0.82, avec un intervalle de confiance à 95% de [0.75 ; 0.86], indiquant une bonne consistance interne des items de l'échelle de Hamilton à ce moment. Elle n'est pas exagérément lourde ni redondante.

3.B.4 Validité externe = l'instrument mesure-t-il ce qu'il est censé mesurer ?

De la même manière qu'à J0, la validité externe de l'échelle de Hamilton à J56 peut être évaluée en examinant la corrélation entre le score total de Hamilton et les dimensions du SCL-90.

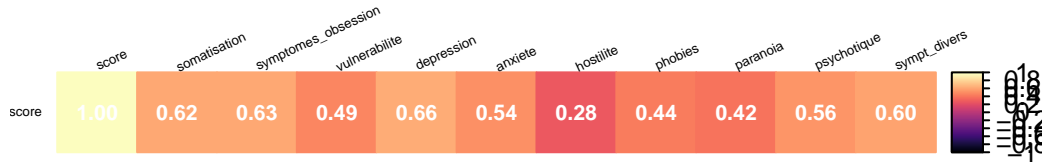


FIGURE 15 – Corrélation entre score total de Hamilton et les dimensions du SCL-90 à J56

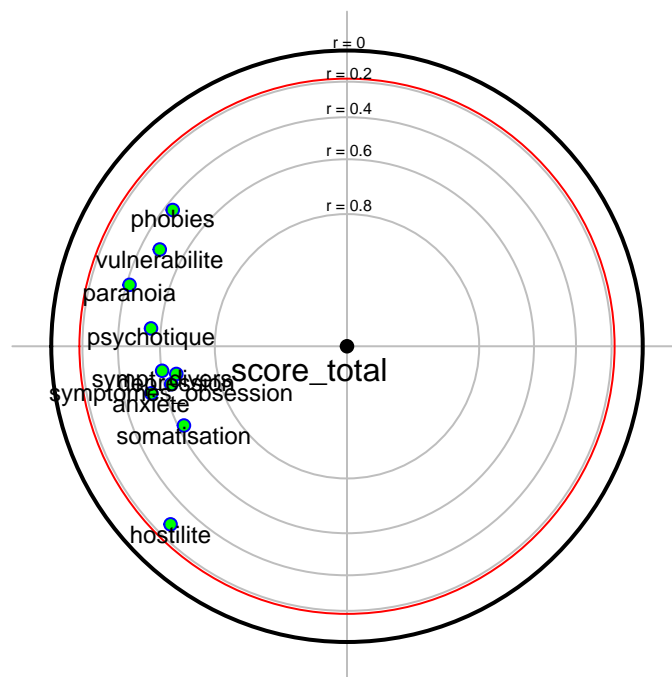


FIGURE 16 – ACP focalisée du score total de l'échelle de Hamilton et des dimensions du SCL-90 à J56

3.B.4.1 Corrélation entre le score de Hamilton et les dimensions du SCL-90 à J56

- Les corrélations sont plus importantes entre les sous-scores de SCL90 et le score d'Hamilton à J56, : ces deux échelles mesurent probablement la même chose.
- La corrélation la plus élevée à J56 est observée avec la dimension « Dépression » du SCL-90 ($r = 0,67$), ce qui renforce l'idée d'une validité convergente accrue à ce temps.

3.B.4.2 Conclusion validité externe à J56 La validité convergente entre le score total de Hamilton et les dimensions du SCL-90 à J56 est modérée à forte, suggérant que l'échelle de Hamilton mesure des aspects pertinents de la détresse psychologique générale à ce moment.

3.B.5 Conclusion globale à J56

À J56, l'échelle de dépression de Hamilton présente une structure dimensionnelle essentiellement unidimensionnelle, avec une bonne consistance interne des items (α de Cronbach = 0.82) et une validité convergente modérée à forte avec les dimensions du SCL-90.

Au total, ces résultats suggèrent que l'échelle de Hamilton est un instrument valide et fiable pour mesurer la dépression dans cette population à J56.

3.B.6 Synthèse des résultats de validation de l'échelle de Hamilton à J0 et J56

Tableau récapitulatif des principaux résultats de validation de l'échelle de Hamilton à J0 et J56 :

Aspect	J0	J56
Structure dimensionnelle	Multidimensionnelle (3 facteurs)	Essentiellement unidimensionnelle
Consistance interne (<i>Alpha</i> de Cronbach)	Faible ($\alpha = 0.45$ [0.39 ; 0.51])	Bonne ($\alpha = 0.82$ [0.75 ; 0.86])
Validité convergente (corrélation avec SCL-90)	Modérée (max = 0.40)	Plus élevée (max = 0.85)
Conclusion globale	Structure dimensionnelle peu claire, faible consistance interne, validité convergente modérée	Essentiellement unidimensionnelle, bonne consistance interne, validité convergente modérée à forte

Note

Validité divergente

La validité divergente (ou discriminante) fait référence à la capacité d'un instrument de mesure à ne pas être fortement corrélé avec des concepts ou des mesures qui sont théoriquement distincts. En d'autres termes, un instrument doit montrer qu'il mesure un concept spécifique sans être influencé de manière significative par d'autres concepts non liés.

En l'occurrence, on ne dispose pas d'instruments mesurant des concepts clairement distincts de la dépression dans ce jeu de données, ce qui limite notre capacité à évaluer la validité divergente de l'échelle de Hamilton dans ce contexte.

4 Question 2 : Comparaison de la réponse au traitement entre deux groupes de patients

Note

Consigne

“A partir du score brut de Hamilton, déterminez si les patients du groupe `groupe=1` répondent mieux au traitement que les patients du groupe `groupe=0`. (utilisez d’abord une approche LOCF (last observation carried forward), puis un modèle mixte).”

4.A Gestion des données

Il faut utiliser le jeu de données `hdrs_groupe`, qui contient les scores bruts de l’échelle de Hamilton à chaque visite (J0, J4, J7, J14, J21, J28, J42, J56) ainsi que le groupe de traitement (`GROUPE`, 0 ou 1) pour chaque patient.

4.B Scores bruts à J0 dans chaque groupe

- Présentation en tableau des scores bruts de l’échelle de Hamilton à J0 dans chaque groupe de traitement avec effectifs, moyennes, écarts-types, médianes et intervalles interquartiles.

TABLE 7 – Scores bruts de l’échelle de Hamilton à J0 par groupe de traitement

GROUPE	N	Moyenne	Écart type	Médiane	Q1	Q3
0	75	28.1	4.43	27	26	30
1	71	27.8	3.97	27	25	30

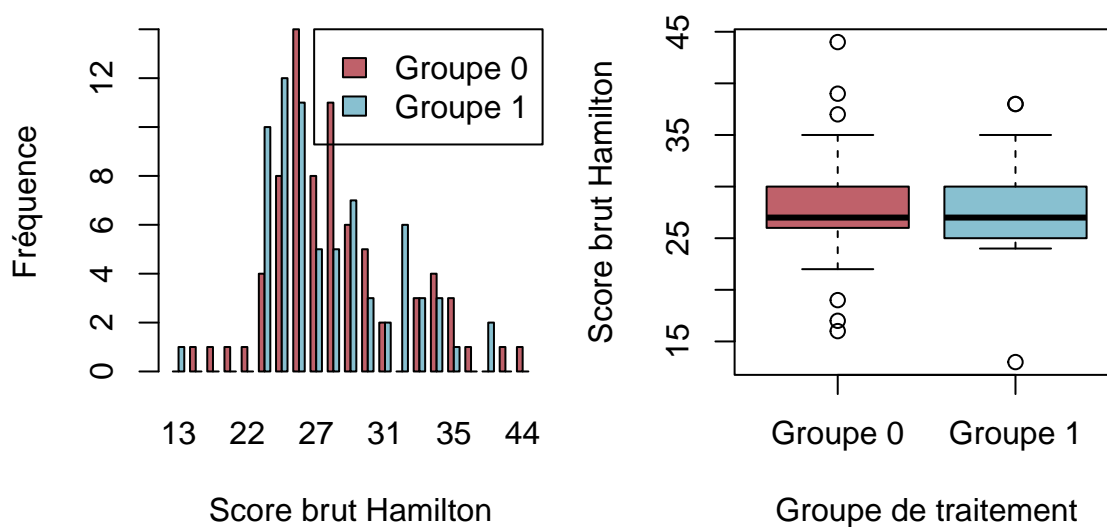


FIGURE 17 – Diagramme en bâtons et boxplot présentant les scores bruts de l’échelle de Hamilton à J0

4.B.1 Comparaison des scores bruts à J0 entre les deux groupes

- Un diagramme en bâtons des scores bruts à J0 à déjà été présenté ci-dessus.
- La distribution semble approximativement normale dans chaque groupe.
- On peut donc comparer les scores bruts moyens entre les deux groupes par un test t de Student pour échantillons indépendants.

Characteristic	0 N = 75 ¹	1 N = 71 ¹	p-value ²
score	28.05 (4.43)	27.79 (3.97)	0.7

¹Mean (SD)

²Two Sample t-test

NB : la fonction utilisée ici est `tbl_summary` du package `gtsummary` permet de réaliser des tableaux de synthèse avec des tests statistiques intégrés.

4.C Évolution des scores bruts entre J0 et J56 dans chaque groupe

4.C.1 Évolution du nombre de patients dans chaque groupe à chaque visite

- Nombre de données entre chaque visite entre J0 et J56 dans chaque groupe :

TABLE 8 – Nombre de patients par groupe de traitement et visite

	J0	J4	J7	J14	J21	J28	J42	J56
0	75	72	71	70	69	68	65	64
1	71	67	65	63	60	59	57	56

A la visite J56, les visites manquent pour 11 patients sur 75 dans le groupe 0 (14,7%), contre 15 patients sur 56 dans le groupe 1 (26,8%).

4.C.2 Évolution des scores bruts moyens entre J0 et J56 dans chaque groupe

On peut représenter :

- L'évolution des scores bruts moyens dans chacun des groupes avec des boxplot à chaque visite

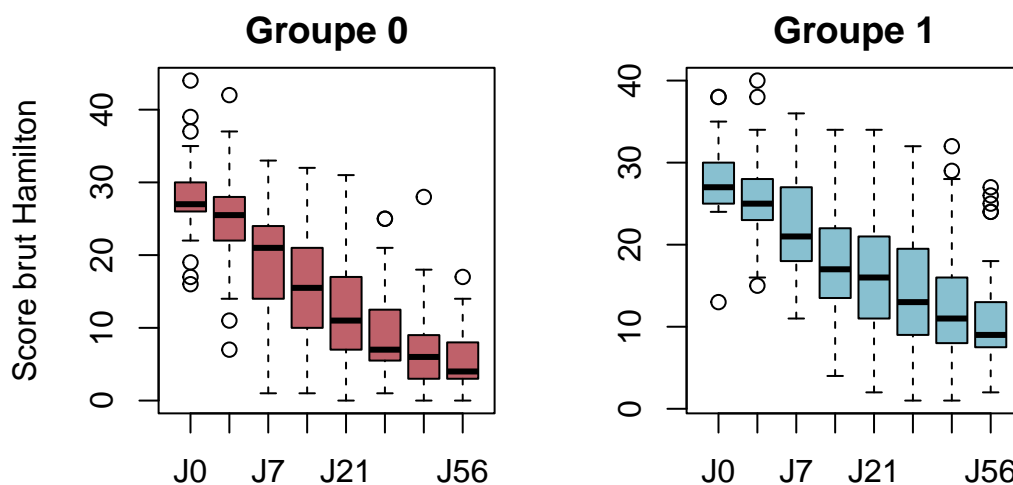


FIGURE 18 – Évolution des scores moyens entre J0 et J56 dans chaque groupe

- L'évolution par patients (dans chacun des groupes), avec un diagramme en spaghetti, d'interprétation plus difficile mais qui montre les trajectoires individuelles.

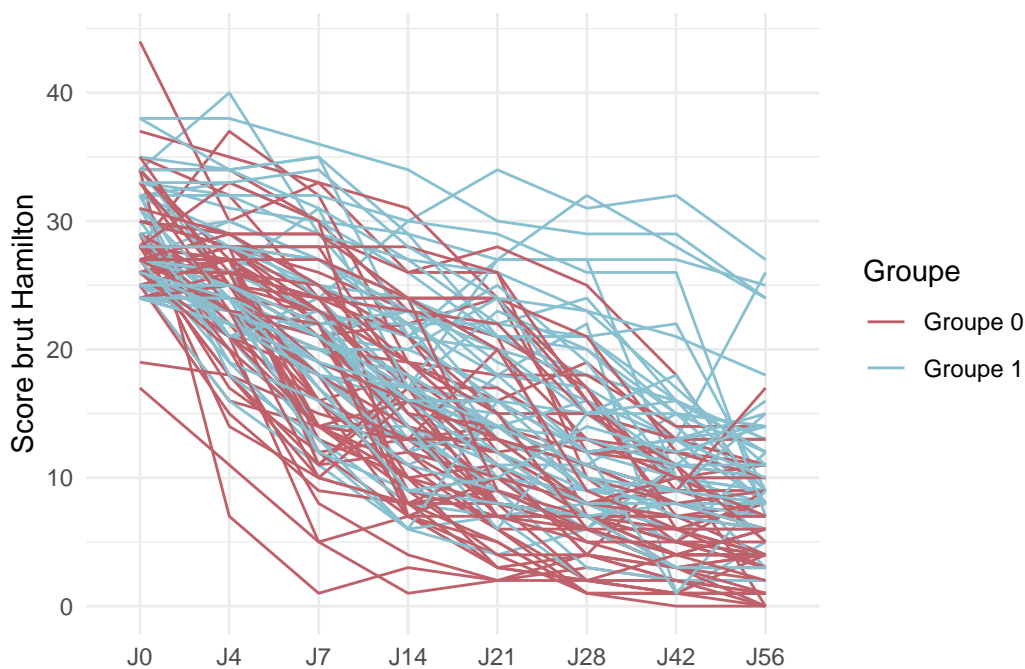


FIGURE 19 – Évolution individuelle des scores entre J0 et J56 dans chaque groupe

- De nouveaux boxplots qui montre le score à J56 côte à côte pour chaque groupe.

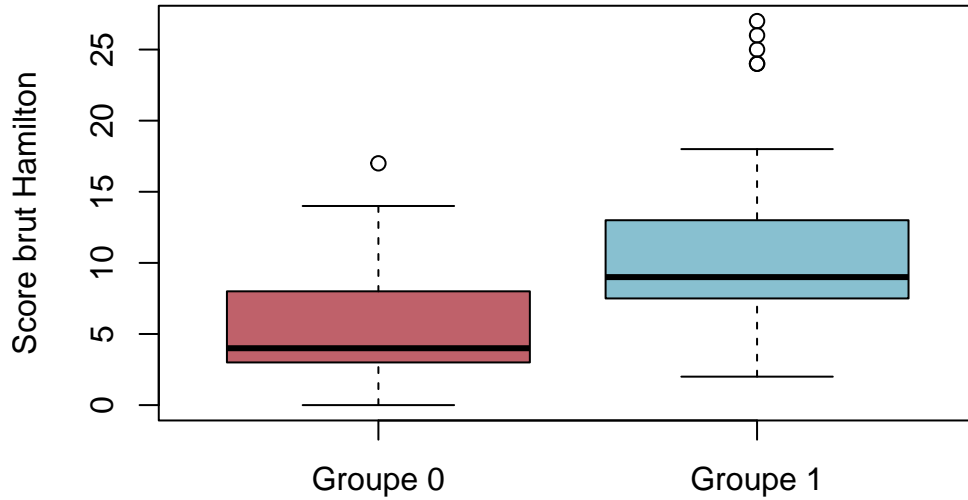


FIGURE 20 – Scores bruts de l’échelle de Hamilton à J56 dans chaque groupe de traitement

Quelle que soit la méthode de visualisation, on observe une tendance diminution des scores bruts moyens de l’échelle de Hamilton entre J0 et J56 dans les deux groupes de traitement.

La diminution semble plus marquée dans le groupe 0 que dans le groupe 1.

4.D Comparaison de la réponse au traitement entre deux groupes de patients

L’objectif est donc de déterminer si un des groupes répond mieux au traitement que l’autre, en comparant l’évolution des scores bruts entre J0 et J56 dans chaque groupe, en utilisant d’abord une approche LOCF (last observation carried forward), puis un modèle mixte.

4.D.1 Comparaison de la réponse en utilisant la méthode LOCF

4.D.1.1 Imputation des données manquantes par la méthode LOCF (last observation carried forward) Méthode LOCF = on remplace les valeurs manquantes par la dernière valeur observée pour chaque patient.

On applique la fonction `na.locf` du package `zoo` pour réaliser cette imputation.

Cette méthode suppose implicitement une stabilité du score après la dernière observation, ce qui peut introduire un biais si l’évolution se poursuit.

TABLE 9 – Nombre de patients par groupe de traitement et visite après imputation LOCF

	J0	J14	J21	J28	J4	J42	J56	J7
0	75	75	75	75	75	75	75	75
1	71	71	71	71	71	71	71	71

Le tableau permet de vérifier que l’imputation LOCF a bien permis d’avoir des données pour tous les patients à chaque visite.

4.D.1.2 Comparaison de la réponse au traitement entre les deux groupes après imputation LOCF Il faut comparer l'évolution des scores bruts moyens entre J0 et J56, donc la différence de score entre J56 et J0, entre les deux groupes.

Pour ça, on crée une nouvelle variable “différence” qui correspond à la différence entre le score à J56 et le score à J0 (après imputation LOCF).

Un test t de Student est utilisé pour comparer la différence moyenne de score entre J0 et J56 entre les deux groupes.

NB : le test t permet de comparer les moyennes de deux échantillons indépendants.

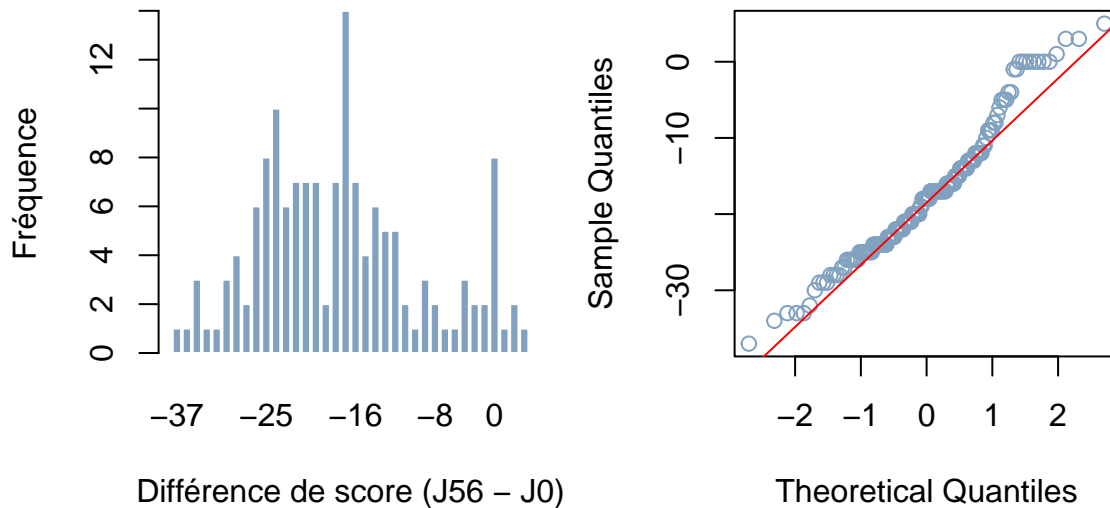


FIGURE 21 – Diagramme en bâtons et QQ-plot de la différence de score entre J0 et J56 (après imputation LOCF) sur l'échantillon complet

Les conditions de validité semblent respectées pour le test t de Student sur la différence de score entre J0 et J56 (après imputation LOCF) sur l'échantillon complet, à savoir : indépendance des échantillons,

Characteristic	0 N = 75 ¹	1 N = 71 ¹	p-value ²
difference	-19.92 (8.68)	-14.79 (8.15)	<0.001

¹Mean (SD)

²Two Sample t-test

Le test t montre une différence moyenne de score entre J0 et J56 significativement plus élevée dans le groupe 0 (-19,92) que dans le groupe 1 (-14,79), avec une p-value < 0.001.

Le problème majeur de cette méthode est qu'elle ne tient pas en compte que les mesures sont répétées chez les mêmes patients !

En utilisant un modèle mixte, on va pouvoir prendre en compte cette dépendance entre les mesures répétées en modélisant un effet aléatoire pour chaque patient.

Le modèle mixte est justifié ici par la structure en mesures répétées, qui induit une corrélation intra-patient non prise en compte par des tests simples.

4.D.2 Comparaison de la réponse en utilisant un modèle mixte

Ici on veut comparer l'évolution du score HDRS entre les groupes **en tenant compte des mesures répétées** (plusieurs visites par patient).

Un modèle linéaire mixte (LMM) est adapté car il modélise la corrélation intra-patient.

De plus, les modèles mixtes permettent d'utiliser toutes les observations disponibles sans imputation explicite des valeurs manquantes, sous l'hypothèse de données manquantes aléatoires.

- Variable à expliquer : **score** (après imputation LOCF), quantitative, discrète.
- Effets fixes : **VISIT** et **GROUPE**.
- Effet aléatoire : un intercept par patient (1|NUMERO) pour tenir compte du niveau moyen différent selon les patients, donc une ordonnée à l'origine différente pour chaque patient.

On peut envisager deux types modèles :

- **Sans interaction** : suppose des trajectoires « parallèles » à chaque visite (différence entre groupes constante au cours du temps).
- **Avec interaction VISIT*GROUPE** : teste si l'évolution au cours du temps diffère entre groupes (c'est le test principal pour « répond mieux »).
- On pourra ensuite évaluer l'intérêt de l'ajout du terme d'interaction dans le modèle.

La question résiduelle est le codage de la variable **VISIT** : on peut la traiter comme une variable catégorielle (facteur) ou continue (temps).

- Il paraît plus représentatif de coder la variable **VISIT** comme une variable continue (temps), car cela permet d'obtenir un coefficient en fonction du temps (jours) et donc d'interpréter plus facilement l'effet du temps sur le score.
- Si on code la variable **VISIT** comme un facteur, on obtient un coefficient pour chaque visite par rapport à la visite de référence (J0), ce qui est moins intuitif.

Dans le code, on fera les deux pour voir si ça change les résultats.

4.D.2.1 Modèle mixte sans interaction

4.D.2.1.1 Modèle mixte sans interaction avec VISIT comme facteur et J0 comme référence

TABLE 10 – Estimation des coefficients du modèle mixte sans interaction entre VISIT (facteur) et GROUPE

Coefficient	Estimate	Std_Error	t_value	CI_lower	CI_upper	p_value
(Intercept)	26.523	0.638	41.555	25.275	27.771	< 0.001
VISITJ4	-2.889	0.498	-5.796	-3.863	-1.914	< 0.001
VISITJ7	-7.142	0.502	-14.223	-8.123	-6.160	< 0.001
VISITJ14	-11.250	0.506	-22.237	-12.238	-10.261	< 0.001
VISITJ21	-13.607	0.511	-26.643	-14.605	-12.609	< 0.001
VISITJ28	-16.298	0.513	-31.759	-17.301	-15.295	< 0.001
VISITJ42	-18.546	0.519	-35.704	-19.561	-17.530	< 0.001
VISITJ56	-20.010	0.522	-38.333	-21.030	-18.990	< 0.001
GROUPE	2.883	0.795	3.627	1.324	4.440	< 0.001

- La t-value et la p-value de l’intercept n’ont pas d’intérêt ici, car le test de Wald pour l’intercept n’a pas de signification pratique, il teste l’hypothèse $H_0 : \text{intercept} = 0$, ce qui n’a pas de sens dans ce contexte.
- Intercept (26.523) : score moyen estimé à J0 dans le groupe de référence (groupe 0).
- Effet du temps (VISIT) : tous les coefficients VISITJx sont négatifs et significatifs → le score diminue significativement par rapport à J0. À J56, la baisse moyenne est de 20.010 points (IC95% [-21.030 ; -18.990]).
 - NB : ici l’effet du temps est fait quel que soit le groupe !
- Effet du groupe (GROUPE = 2.883) : à visite donnée, le groupe 1 a en moyenne +2.883 points de score par rapport au groupe 0 (IC95% [1.324 ; 4.440], $p = 0.0004$), donc des scores plus élevés en moyenne sur l’ensemble du suivi.

Ce modèle décrit une amélioration globale au cours du temps et un décalage moyen entre groupes, mais il ne permet pas de conclure sur une différence de réponse “en pente” entre groupes (il faut le modèle avec interaction $\text{VISIT} \times \text{GROUPE}$ pour ça)

De plus, le fait d’avoir mis **VISIT** comme un facteur empêche d’interpréter facilement l’effet du temps sur le score. Il faut interpréter comme la différence de score entre chaque visite et J0, ce qui est moins intuitif.

4.D.2.1.2 Modèle mixte sans interaction avec VISIT comme variable continue

TABLE 11 – Estimation des coefficients du modèle mixte sans interaction entre VISIT (continue) et GROUPE

Coefficient	Estimate	Std_Error	t_value	CI_lower	CI_upper	p_value
(Intercept)	22.968	0.581	39.543	21.831	24.106	< 0.001
VISIT_num	-0.352	0.009	-40.792	-0.369	-0.335	< 0.001
GROUPE	2.997	0.799	3.752	1.431	4.562	< 0.001

- Intercept = 22.968 : score HDRS prédit pour le groupe de référence (groupe 0) quand VISIT_num = 0 (jour 0), selon le modèle linéaire en jours. Ce n’est pas forcément égal à la moyenne empirique à J0.
- NB : l’intercept correspond donc ici à la valeur prédite par la droite ajustée sur toutes les visites !
- VISIT_num = -0.352 : le score diminue en moyenne de 0.352 point par jour (IC95% [-0.369 ; -0.335], $p < 0.001$). Sur 56 jours, ça correspond à une baisse attendue d’environ $0.352 \times 56 = 19.7$ points.
- GROUPE = +2.997 : à temps donné (même jour), le groupe 1 a un score moyen plus élevé d’environ 3 points que le groupe 0 (IC95% [1.431 ; 4.562], $p = 0.0003$). Sans interaction, cette différence est supposée constante dans le temps.

Le score baisse avec le temps (modélisé linéairement en jours) et le groupe 1 reste en moyenne plus haut que le groupe 0 ; ce modèle ne teste pas une différence d’évolution entre groupes (il faudrait ajouter l’interaction $\text{VISIT_num} * \text{GROUPE}$).

4.D.2.2 Modèle mixte avec interaction

4.D.2.2.1 Modèle mixte avec interaction avec VISIT comme facteur et J0 comme référence

TABLE 12 – Estimation des coefficients du modèle mixte avec interaction entre VISIT (facteur) et GROUPE

Coefficient	Estimate	Std_Error	t_value	p_value
(Intercept)	28.053	0.699	40.115	< 0.001
VISITJ4	-3.364	0.674	-4.990	< 0.001
VISITJ7	-8.663	0.677	-12.790	< 0.001
VISITJ14	-12.587	0.680	-18.497	< 0.001
VISITJ21	-15.640	0.683	-22.882	< 0.001
VISITJ28	-18.875	0.686	-27.495	< 0.001
VISITJ42	-21.328	0.696	-30.654	< 0.001
VISITJ56	-22.307	0.699	-31.914	< 0.001
GROUPE	-0.265	1.003	-0.264	0.7920
VISITJ4 :GROUPE	0.949	0.969	0.979	0.3278
VISITJ7 :GROUPE	3.141	0.977	3.216	< 0.01
VISITJ14 :GROUPE	2.778	0.984	2.822	< 0.01
VISITJ21 :GROUPE	4.289	0.994	4.313	< 0.001
VISITJ28 :GROUPE	5.464	0.999	5.469	< 0.001
VISITJ42 :GROUPE	5.889	1.011	5.824	< 0.001
VISITJ56 :GROUPE	4.853	1.016	4.775	< 0.001

- L'interprétation est identique à celle du modèle sans interaction pour les effets principaux (Intercept, VISIT, GROUPE).
- Pour interpréter le terme d'interaction, il faut regarder les coefficients des interactions VISITJx :GROUPE1.
- Ici, tous les coefficients d'interaction VISITJx :GROUPE1 sont positifs et significatifs à partir de J7, ce qui indique que la diminution du score HDRS est significativement moins marquée dans le groupe 1 que dans le groupe 0 à partir de la visite J7 par rapport à la visite J0.

4.D.2.2.2 Modèle mixte avec interaction avec VISIT comme variable continue

TABLE 13 – Estimation des coefficients du modèle mixte avec interaction entre VISIT (continue) et GROUPE

Coefficient	Estimate	Std_Error	t_value	p_value
(Intercept)	23.807	0.600	39.669	< 0.001
VISIT_num	-0.394	0.012	-33.792	< 0.001
GROUPE	1.251	0.863	1.450	0.1486
VISIT_num :GROUPE	0.090	0.017	5.283	< 0.001

- L'interprétation est identique à celle du modèle sans interaction pour les effets principaux (Intercept, VISIT_num, GROUPE).
- Le coefficient d'interaction VISIT_num :GROUPE est de 0.090 ($p < 0.001$). Ce coefficient représente la différence de pente de l'évolution du score HDRS entre le groupe 1 et le groupe 0 (qui est le groupe de référence).

- Étant donné que le coefficient est positif, la pente pour le groupe 1 est moins négative, ce qui indique une amélioration plus lente.
- La pente pour le groupe 0 est de -0.394 (le score diminue en moyenne de 0.394 points par jour).
- La pente pour le groupe 1 est de $-0.394 + 0.090 = -0.304$ (le score diminue en moyenne de seulement 0.304 points par jour).
- En conclusion, l'amélioration du score HDRS est significativement **plus lente** dans le groupe 1 que dans le groupe 0.

4.D.2.2.3 Test de l'effet d'interaction par rapport au modèle sans interaction Pour tester l'effet d'interaction, on réalise un test F pour comparer l'effet de l'ajout d'un terme d'interaction entre VISIT et GROUPE par rapport au modèle sans interaction à l'aide de la fonction `drop1`.

TABLE 14 – Test F de l'effet d'interaction entre VISIT (continue) et GROUPE

Term	NumDF	DenDF	F_value	p_value
VISIT_num :GROUPE	1	924.7691	27.908	< 0.001

La p-value associée au test de l'effet d'interaction est < 0.001 , ce qui indique que l'ajout du terme d'interaction entre VISIT et GROUPE améliore significativement le modèle par rapport au modèle sans interaction.

En pratique, cela signifie que l'évolution des scores HDRS au cours du temps diffère significativement entre les deux groupes de traitement.

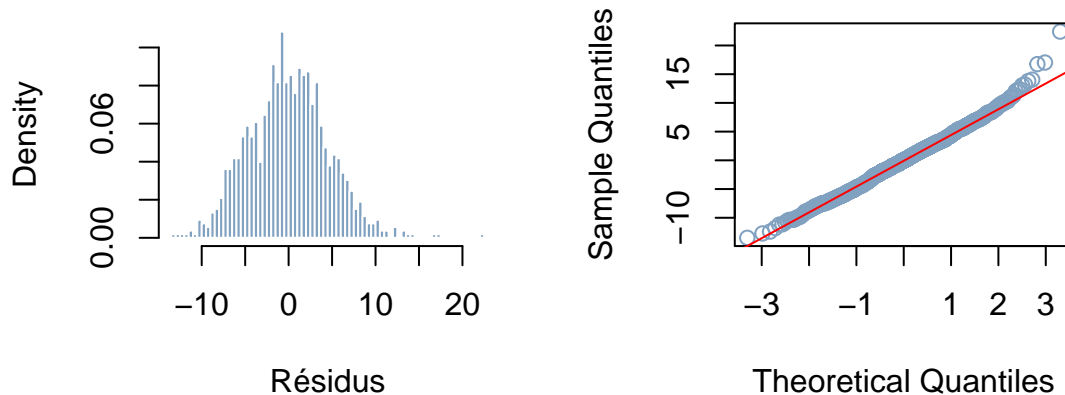


FIGURE 22 – Normalité des résidus du modèle mixte avec interaction entre VISIT (continue) et GROUPE

Les résidus semblent suivre une distribution normale, ce qui indique que les conditions de validité du modèle mixte sont respectées (avec VISIT codé comme une variable continue et un terme d'interaction entre VISIT et GROUPE).

4.E Conclusion sur la comparaison de la réponse au traitement entre les deux groupes de patients

- La méthode LOCF suivie d'un test t de Student montre une différence moyenne de score entre J0 et J56 significativement plus élevée dans le groupe 0 que dans le groupe 1 ($p < 0.001$).
- Le modèle mixte avec interaction entre VISIT et GROUPE confirme que l'évolution des scores HDRS au cours du temps diffère significativement entre les deux groupes de traitement ($p < 0.001$ pour l'effet d'interaction).
- Une interaction significative groupe \times temps indique des trajectoires d'évolution différentes entre groupes, et pas seulement un décalage moyen constant.

5 Question 3 : Analyse de survie

Note

Consigne

“On répond aussi parfois à cette question en considérant le critère binaire censuré « réponse au traitement » défini par une chute de 50% à l’échelle de Hamilton par rapport à J0. En utilisant ce nouveau critère, répondez donc de nouveau à la question 2.”

5.A Critère binaire de réponse (50% à la dernière visite observée)

L’objectif est de définir un critère binaire de réponse au traitement basé sur une chute d’au moins 50% du score HDRS par rapport à J0, en utilisant la dernière visite observée pour chaque patient.

Les étapes sont :

- Obtention du score HDRS de la dernière visite observée (non manquante) pour chaque patient.
- Calcul de la chute par rapport à J0.
- Si la chute est d’au moins 50%, le patient est considéré comme ayant répondu au traitement (réponse = 1), sinon non (réponse = 0).
- Fusion de cette information avec les données au format long `hdrs_groupe`, dans une variable `reponse` (et un nouveau data.frame `hdrs_groupe_reponse`).

5.B Intégration du critère binaire de réponse dans une analyse de survie

L’objectif est d’utiliser le critère binaire de réponse (chute d’au moins 50% du score HDRS par rapport à J0) dans une analyse de survie.

Il faut donc définir :

- Un temps de suivi pour chaque patient, qui correspond au délai jusqu’à la **première** réponse observée (chute $\geq 50\%$ par rapport à J0). Si une visite est manquante, on considère une censure à la visite précédente (logique du PDF DEGACHI), sinon censure administrative à J56.
- Un indicateur d’événement, qui vaut 1 si la réponse est observée et 0 si le patient est censuré.

Il y a une censure administrative à 56 jours, et une censure au premier manquant quand une visite est absente.

TABLE 15 – Temps de suivi (en jours) par groupe de traitement

	0	4	7	14	21	28	42	56
0	3	3	16	17	13	12	9	2
1	4	3	9	12	10	9	10	14

5.C Analyse de survie avec le critère binaire de réponse

L’analyse de survie permet de comparer la probabilité de réponse au traitement entre les deux groupes en tenant compte de la censure, ainsi que le délai de survenue de la réponse.

Trois issues sont possibles pour chaque patients :

- Réponse observée (événement = 1) : chute 50% du score HDRS par rapport à J0 à une certaine visite.
- Pas de réponse observée, fin du suivi à 56 jours (événement = 0, temps_suivi = 56).
- Pas de réponse observée, censure au premier manquant (événement = 0, temps_suivi = jour de la première visite manquante).

On représente les courbes de survie en utilisant la méthode de Kaplan-Meier.

Les courbes sont ensuite comparées entre les deux groupes à l'aide d'un test de log-rank, permettant la comparaison des réponses entre les deux groupes.

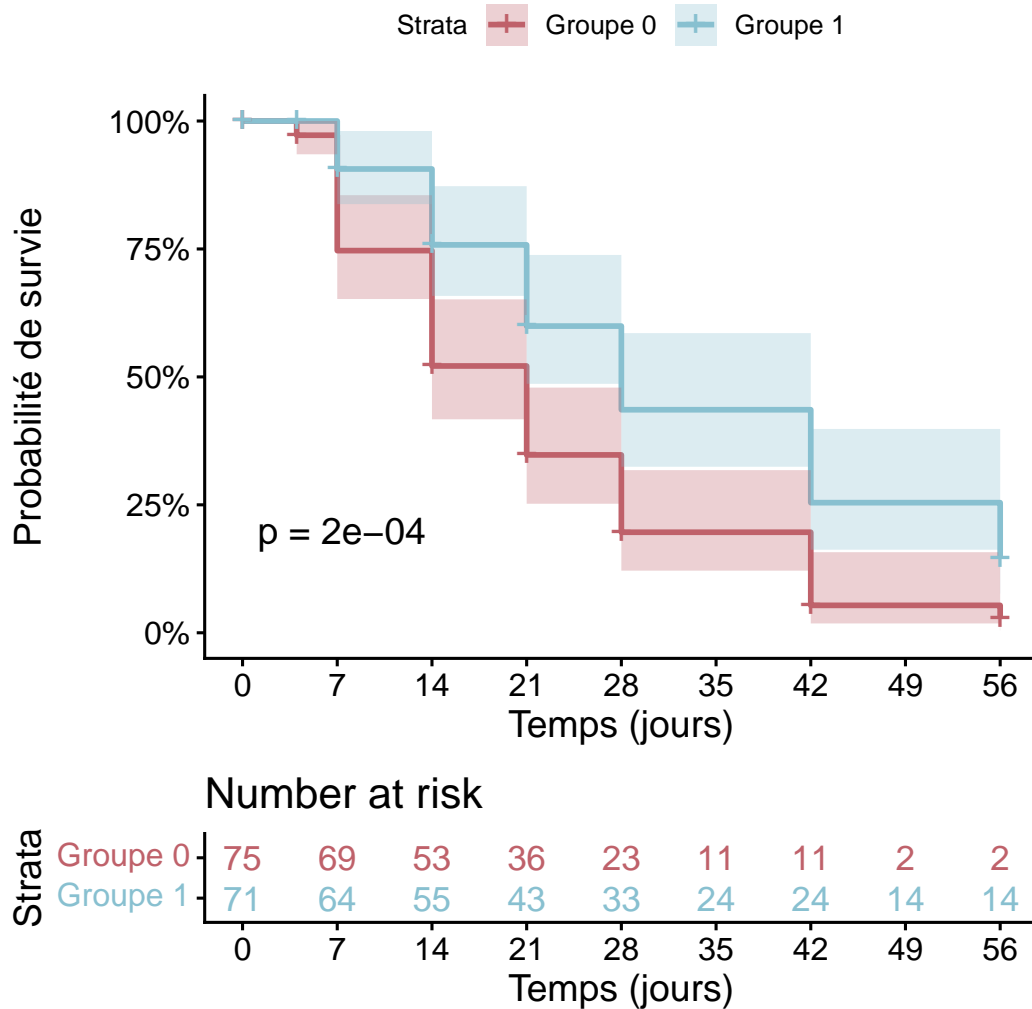


FIGURE 23 – Courbes de survie de la réponse au traitement par groupe de traitement

- Résultat du test de log-rank : **différence significative entre les deux groupes de traitement** ($p < 0.01$), indiquant que la probabilité de réponse au traitement diffère entre les groupes.
- Mais le test de log-rank permet seulement une comparaison globale des courbes de survie entre les groupes.
- Pour évaluer quantitativement l'effet du groupe de traitement sur le temps jusqu'à la réponse,

on peut ajuster un modèle de Cox.

- Il est nécessaire de vérifier l'hypothèse des risques proportionnels avant d'interpréter les résultats du modèle de Cox, à l'aide d'un diagramme des risques instantanés. Ici, le diagramme est linéaire, ce qui suggère que l'hypothèse des risques proportionnels est raisonnablement respectée.

TABLE 16 – Estimation des coefficients du modèle de Cox pour le critère binaire de réponse au traitement

Coefficient	Estimate	HR	CI_lower	CI_upper	p_value
GROUPE1	-0.714	0.49	0.334	0.718	3e-04

Risques proportionnels – Modèle de Cox binaire

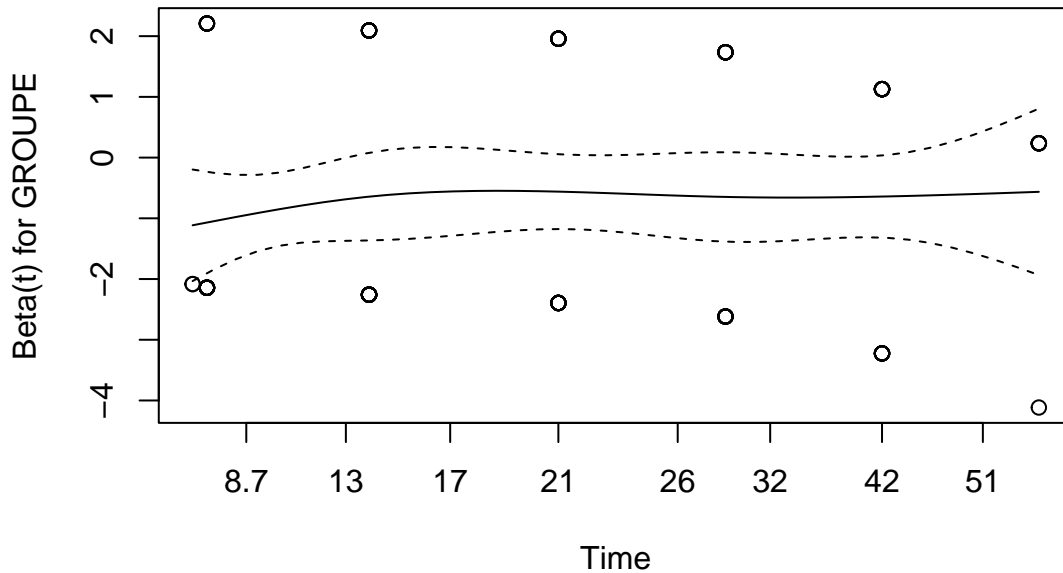


FIGURE 24 – Test des risques proportionnels du modèle de Cox binaire

Le modèle de Cox retrouve un $HR = 0.49$ avec une $p\text{-value}$ significative $< 1e-03$, indiquant que le risque instantané de réponse au traitement est réduit de 51% dans le groupe 1 par rapport au groupe 0.

On peut donc conclure que les patients du groupe 0 répondent mieux au traitement que ceux du groupe 1, en utilisant le critère binaire de réponse basé sur une chute de 50% du score HDRS par rapport à J0.

6 Conclusion générale

- L'échelle Hamilton est une échelle d'hétéro-évaluation utilisée pour évaluer la sévérité de la dépression.
- L'échelle a une mauvaise cohérence interne initialement, puis devient fiable avec une structure stable au cours du suivi.

Elle est comparable à d'autres instruments admis, par exemple l'échelle SCL90.

La comparaison entre les deux groupes de sujets étudiés par l'échelle Hamilton à l'aide :

- d'une comparaison avec une approche LOCF, utilisant une régression linéaire
- de modèles mixtes sans et avec interaction entre le groupe et la durée de suivi
- d'une analyse de survie

retrouve dans tous les cas une **différence significative** entre les deux groupes, avec une amélioration plus marquée et plus rapide chez les patients du groupe 0.

7 Annexe – Code R

```
library(forecast)
library(plotrix)
library(randomForest)
library(tidyr)
library(epiR)
library(viridisLite)
library(ggplot2)
library(binom)
library(survminer)
library(pROC)
library(treemap)
library(psy)
library(MASS)
library(rpart)
library(rpart.plot)
library(plotly)
library(lmerTest)
library(psych)
library(lme4)
library(prettyR)
library(zoo)
library(kableExtra)
library(gtsummary)
library(dplyr)
library(lattice)
library(survey)
library(corrplot)
library(mice)
library(paletter)
library(nord)
library(wesanderson)
library(qgraph)
library(nlme)
library(pwr)
library(ape)
library(survival)
library(gmodels)
library(httptest)
library(e1071)
library(psy)
library(reshape2)
knitr::opts_chunk$set(echo = TRUE)

#file_qmd <- "/Users/thomashusson/Documents/Projets/M2biostatistiques/devoir_stats_avancees/
#devoir_stats_avancees.qmd"

#extraction du code R pour le mettre à la fin du quarto md
file_qmd <- knitr::current_input()
if (is.null(file_qmd) || file_qmd == "") {
  file_qmd <- "devoir_stats_avancees.qmd"
}
if (!file.exists(file_qmd)) {
  stop(paste("Fichier Quarto introuvable:", file_qmd))
}
lines <- readLines(file_qmd, warn = FALSE)

# trouver les debuts de chunks R
starts <- grep("````\\{r", lines)
# trouver les fins de chunks
ends <- grep("````\\s*$", lines)

if (length(starts) != length(ends)) {
  stop("Extraction des chunks R impossible : démarcations incohérentes")
}

code_r <- character(0)
for (i in seq_along(starts)) {
  # lignes à l'intérieur du chunk (excluant les balises de début et fin)
```

```

    chunk_lines <- lines[(starts[i] + 1):(ends[i] - 1)]
    # filtrer : garder seulement les lignes qui **ne sont pas des options Quarto**
    real_code <- chunk_lines[!grepl("^\\s*#\\s*", chunk_lines)]
    # ajouter au vecteur final
    code_r <- c(code_r, real_code, "")
  }

outfile <- file.path(dirname(file_qmd), "all_chunks_code.R")
cat(code_r, file = outfile, sep = "\n")

library(readxl)
scl90 <- read_excel("outils_autoeval.xls")
groupe <- read_excel("outils_groupe.xls")
hdrs <- read_excel("outils_hdrs.xls")

describe(scl90)
summary(scl90)

# ordonne chronologiquement les visites pour chaque patient
scl90$VISIT <- factor(scl90$VISIT,
  levels = c("J0", "J4", "J7", "J14", "J21", "J28", "J42", "J56"),
  ordered = TRUE)

# ordonner les visites en fonction du numéro de patient
scl90 <- scl90[order(scl90$NUMERO, scl90$VISIT), ]

# identification des doublons
scl90$NUMERO[duplicated(scl90)]

# nombre de patients uniques
length(unique(scl90$NUMERO))

# visualisation des données aberrantes
apply(scl90, 2, table, useNA = "always")
scl90[, 3:92][scl90[, 3:92] > 4] <- NA
apply(scl90, 2, table, useNA = "always")

# données manquantes
sum(is.na(scl90))
# proportion :
sum(is.na(scl90)) / (nrow(scl90) * ncol(scl90))*100

#imputation par le mode pour chaque question
scl_questions <- c(names(select(scl90,starts_with("Q"))))
for (question in scl_questions) {
  original <- scl90[[question]]
  factorized <- as.factor(original)
  mode_value <- as.integer(names(which.max(table(factorized))))
  imputed <- original
  imputed[is.na(imputed)] <- mode_value
  scl90[[question]] <- imputed
}
# vérifier qu'il n'y a plus de NA
sum(is.na(scl90))

# création des scores moyens par dimension
scl90$somatisation <- apply(scl90[,c(3,6,14,29,44,50,51,54,55,58,60,42)],1,mean,na.rm=TRUE)
scl90$symptomes_obsession <- apply(scl90[,c(11,12,30,40,5,47,48,53,57,67)],1,mean,na.rm=TRUE)
scl90$vulnerabilite <- apply(scl90[,c(8,23,36,38,39,43,63,71,75)],1,mean,na.rm=TRUE)
scl90$depression <- apply(scl90[,c(7,16,17,22,24,28,31,32,33,34,56,73,81)],1,mean,na.rm=TRUE)
scl90$anxiete <- apply(scl90[,c(4,19,25,35,41,59,74,80,82,88)],1,mean,na.rm=TRUE)
scl90$hostilite <- apply(scl90[,c(13,26,65,69,76,83)],1,mean,na.rm=TRUE)
scl90$phobies <- apply(scl90[,c(15,27,49,72,77,84,52)],1,mean,na.rm=TRUE)
scl90$paranoia <- apply(scl90[,c(10,20,45,70,78,85)],1,mean,na.rm=TRUE)
scl90$psychotique <- apply(scl90[,c(9,18,37,64,79,86,87,89,92,90)],1,mean,na.rm=TRUE)
scl90$sympt_divers <- apply(scl90[,c(21,46,61,62,66,68,91)],1,mean,na.rm=TRUE)

# création d'un nouveau dataframe avec uniquement les 10 dimensions
dimensions <- c("somatisation", "symptomes_obsession", "vulnerabilite", "depression", "anxiete",
  "hostilite", "phobies", "paranoia", "psychotique", "sympt_divers")
scl90_dim <- scl90[, c("NUMERO", "VISIT", dimensions)]

hdrs$VISIT <- factor(hdrs$VISIT,
  levels = c("J0", "J4", "J7", "J14", "J21", "J28", "J42", "J56"),

```

```

        ordered = TRUE)

# ordonner les visites en fonction du numéro de patient
hdrs <- hdrs[order(hdrs$NUMERO, hdrs$VISIT), ]

# identification des doublons
hdrs$NUMERO[duplicated(hdrs)]

# nombre de patients uniques
length(unique(hdrs$NUMERO))

# fusion des variables HAMD16A et HAMD16B en une seule variable HAMD16
hdrs$HAMD16 <- ifelse(!is.na(hdrs$HAMD16A), hdrs$HAMD16A, hdrs$HAMD16B)
table(hdrs$HAMD16, useNA = "ifany")

# calcul du score total HDRS
items <- c("HAMD1", "HAMD2", "HAMD3", "HAMD4", "HAMD5", "HAMD6",
          "HAMD7", "HAMD8", "HAMD9", "HAMD10", "HAMD11", "HAMD12",
          "HAMD13", "HAMD14", "HAMD15", "HAMD16", "HAMD17")

#renommer la colonne hdrs$HAMD1 en hdrs$humeur_depressive
colnames(hdrs)[colnames(hdrs) == "HAMD1"] <- "1_humeur_depressive"
colnames(hdrs)[colnames(hdrs) == "HAMD2"] <- "2_sentim_culpabilite"
colnames(hdrs)[colnames(hdrs) == "HAMD3"] <- "3_suicide"
colnames(hdrs)[colnames(hdrs) == "HAMD4"] <- "4_insomnie_debut"
colnames(hdrs)[colnames(hdrs) == "HAMD5"] <- "5_insomnie_milieu"
colnames(hdrs)[colnames(hdrs) == "HAMD6"] <- "6_insomnie_matin"
colnames(hdrs)[colnames(hdrs) == "HAMD7"] <- "7_travail_activite"
colnames(hdrs)[colnames(hdrs) == "HAMD8"] <- "8_ralentissement"
colnames(hdrs)[colnames(hdrs) == "HAMD9"] <- "9_agitation"
colnames(hdrs)[colnames(hdrs) == "HAMD10"] <- "10_anxiete_psychique"
colnames(hdrs)[colnames(hdrs) == "HAMD11"] <- "11_anxiete_somatique"
colnames(hdrs)[colnames(hdrs) == "HAMD12"] <- "12_symptomes_gastro"
colnames(hdrs)[colnames(hdrs) == "HAMD13"] <- "13_symptomes_generaux"
colnames(hdrs)[colnames(hdrs) == "HAMD14"] <- "14_symptomes_genitiaux"
colnames(hdrs)[colnames(hdrs) == "HAMD15"] <- "15_hypochondrie"
colnames(hdrs)[colnames(hdrs) == "HAMD16"] <- "16_perte_poids"
colnames(hdrs)[colnames(hdrs) == "HAMD17"] <- "17_prise_conscience"

# calcul du score total HDRS
hdrs_items <- c("1_humeur_depressive", "2_sentim_culpabilite", "3_suicide", "4_insomnie_debut", "5_
  insomnie_milieu", "6_insomnie_matin", "7_travail_activite", "8_ralentissement", "9_agitation", "10_
  anxiete_psychique", "11_anxiete_somatique", "12_symptomes_gastro", "13_symptomes_generaux", "14_
  symptomes_genitiaux", "15_hypochondrie", "16_perte_poids", "17_prise_conscience")

hdrs$score <- rowSums(hdrs[, hdrs_items], na.rm = TRUE)

# supprimer la ligne 741 de hdrs car les données sont manquantes
hdrs <- hdrs[-741, ]

# supprimer HAMD16A et HAMD16B si présentes
hdrs <- hdrs[, setdiff(names(hdrs), c("HAMD16A", "HAMD16B"))]

summary(groupe)
describe(groupe)
groupe$NUMERO[duplicated(groupe)]
length(unique(groupe$NUMERO))

# ordonner en fonction du numéro de patient
groupe <- groupe[order(groupe$NUMERO), ] # Questions 1 et 2

# fusion des 3 dataframes
hdrs_groupe <- merge(hdrs, groupe, by = "NUMERO", all.x = TRUE)
scl90_groupe <- merge(scl90, groupe, by = "NUMERO", all.x = TRUE)
df_total_wide <- merge(hdrs_groupe, scl90, by = c("NUMERO", "VISIT"), all.x = TRUE)

library(reshape2)

## HDRS -> wide
hdrs_wide <- reshape2::dcast(
  hdrs_groupe,
  NUMERO + GROUPE ~ VISIT,
  value.var = "score"
)

```

```

hdrs_long <- melt(
  hdrs_groupe,
  id.vars = c("NUMERO", "VISIT", "GROUPE"),
  variable.name = "item",
  value.name = "value"
)

## SCL90 -> long
scl90_long <- melt(
  scl90_groupe,
  id.vars = c("NUMERO", "VISIT", "GROUPE"),
  variable.name = "item",
  value.name = "value"
)

## TOTAL (HDRS + SCL90) -> long
df_total_long <- melt(
  df_total_wide,
  id.vars = c("NUMERO", "VISIT", "GROUPE"),
  variable.name = "item",
  value.name = "value"
)

hdrs_J0 <- subset(hdrs_groupe, VISIT == "J0")

# Vraie palette Nord (package nord)
# On prend une palette qualitative (frost) et on l'étend à 5 couleurs
cols_nord <- nord::nord("frost", 5)

# Première série de graphiques (9 au maximum)
par(mfrow = c(3, 3), mar = c(2, 2, 2, 1))
items1 <- hdrs_items[1:min(9, length(hdrs_items))]
for (item in items1) {
  val <- na.omit(hdrs_J0[[item]])
  if (length(val) > 0) {
    m <- max(val)
    counts <- table(factor(val, levels = 0:m))
    barplot(counts,
      main = item,
      xlab = "Score",
      ylab = "Fréquence",
      col = cols_nord[1:(m + 1)],
      border = "white")
  } else {
    plot.new()
    title(main = paste(item, "(pas de données)"))
  }
}
par(mfrow = c(1, 1))

# Deuxième série de graphiques (8 au maximum, de 10 à 17)
if (length(hdrs_items) > 9) {
  par(mfrow = c(3, 3), mar = c(2, 2, 2, 1))
  items2 <- hdrs_items[10:min(17, length(hdrs_items))]
  for (item in items2) {
    val <- na.omit(hdrs_J0[[item]])
    if (length(val) > 0) {
      m <- max(val)
      counts <- table(factor(val, levels = 0:m))
      barplot(counts,
        main = item,
        xlab = "Score",
        ylab = "Fréquence",
        col = cols_nord[1:(m + 1)],
        border = "white")
    } else {
      plot.new()
      title(main = paste(item, "(pas de données)"))
    }
  }
}
par(mfrow = c(1, 1))

```

```

}

hdrs_J0_matrix <- hdrs_J0[, hdrs_items]
corr_matrix_J0 <- cor(hdrs_J0_matrix, use = "pairwise.complete.obs")

corrplot(corr_matrix_J0,
          method = "color",
          type = "upper",
          tl.col = "black",
          addCoef.col = "white",
          number.cex = 0.35,
          tl.cex = 0.6,
          tl.srt = 45,
          col = viridis::plasma(100)
)

#renommer les noms des variables dans un nouveau df copié pour éviter la superposition
hdrs_J0_PCA <- hdrs_J0[,c(hdrs_items)]
colnames(hdrs_J0_PCA) <- c("1","2","3","4","5","6",
                          "7","8","9","10","11","12",
                          "13","14","15","16","17")

mdspca(hdrs_J0_PCA)

head(hdrs_J0[,c(hdrs_items)])
scree.plot(hdrs_J0[,c(hdrs_items)], simu=20, use = "P")

af_J0 <- factanal(
  na.omit(hdrs_J0[hdrs_items]),
  factors = 3,
  rotation = "varimax"
)

# Extraction propre des loadings
loadings_df <- as.data.frame(unclass(af_J0$loadings))

# Création explicite de la colonne Variable à partir des rownames
loadings_df$Variable <- rownames(loadings_df)

# Suppression des rownames pour éviter toute ambiguïté
rownames(loadings_df) <- NULL

# Réorganisation : Variable en première colonne
loadings_df <- loadings_df[, c("Variable", colnames(loadings_df)[1:3])]

# Arrondi
loadings_df[, -1] <- round(loadings_df[, -1], 3)

# Affichage LaTeX
knitr::kable(
  loadings_df,
  caption = "Contribution des facteurs à la variance de la réponse à chaque item du score de
             Hamilton évalué à J0 (analyse factorielle avec rotation varimax à 3 facteurs)",
  booktabs = TRUE,
  align = "lccc"
)

hdrs_J0$f1_somatique <- rowMeans(hdrs_J0[,c("8_ralentissement","12_symptomes_gastro","14_
symptomes_genitiaux","16_perte_poids")], na.rm=TRUE)
hdrs_J0$f2_asthenie <- rowMeans(hdrs_J0[,c("4_insomnie_debut","13_symptomes_generaux")], na.rm=
TRUE)
hdrs_J0$f3_depressif <- rowMeans(hdrs_J0[,c("1_humeur_depressive","2_sentim_culpabilite","3_
suicide","5_insomnie_milieu","6_insomnie_matin","7_travail_activite","9_agitation","10_
anxiete_psychique","11_anxiete_somatique","15_hypochondrie","17_prise_conscience")], na.rm=
TRUE)

hdrs_J0_subscores <- hdrs_J0[,c("f1_somatique","f2_asthenie","f3_depressif")]
colnames(hdrs_J0_subscores) <- c("F1_somatique","F2_asthenie","F3_depressif")
mdspca(hdrs_J0_subscores)

df_fpca <- data.frame(
  score_total = hdrs_J0$score,
  hdrs_J0_subscores
)

fpca(

```



```

    score_total ~ .,
    data = df_fPCA
)

cronbach(hdrs_J0[,hdrs_items])
cronbach(hdrs_J0[,c("8_ralentissement", "12_symptomes_gastro", "14_symptomes_genitiaux", "16_perte_poids")])
cronbach(hdrs_J0[,c("4_insomnie_debut", "13_symptomes_generaux")])
cronbach(hdrs_J0[,c("1_humeur_depressive", "2_sentim_culpabilite", "3_suicide", "5_insomnie_milieu", "6_insomnie_matin", "7_travail_activite", "9_agitation", "10_anxiete_psychique", "11_anxiete_somatique", "15_hypochondrie", "17_prise_conscience")])

#estimation des IC par bootstrap
set.seed(123)
alpha_bootstrap <- function(data, indices) {
  d <- data[indices, ]
  return(cronbach(d)$alpha)
}
library(boot)
# Alpha global
boot_alpha_global <- boot(hdrs_J0[,hdrs_items], alpha_bootstrap, R = 1000)
boot.ci(boot_alpha_global, type = "bca")
# Facteur 1
boot_alpha_f1 <- boot(hdrs_J0[,c("8_ralentissement", "12_symptomes_gastro", "14_symptomes_genitiaux", "16_perte_poids")], alpha_bootstrap, R = 1000)
boot.ci(boot_alpha_f1, type = "bca")
# Facteur 2
boot_alpha_f2 <- boot(hdrs_J0[,c("4_insomnie_debut", "13_symptomes_generaux")], alpha_bootstrap, R = 1000)
boot.ci(boot_alpha_f2, type = "bca")
# Facteur 3
boot_alpha_f3 <- boot(hdrs_J0[,c("1_humeur_depressive", "2_sentim_culpabilite", "3_suicide", "5_insomnie_milieu", "6_insomnie_matin", "7_travail_activite", "9_agitation", "10_anxiete_psychique", "11_anxiete_somatique", "15_hypochondrie", "17_prise_conscience")], alpha_bootstrap, R = 1000)
boot.ci(boot_alpha_f3, type = "bca")

#représentation des alpha de Cronbach avec IC en tableau
alpha_df <- data.frame(
  Scale = c("Global", "F1_somatique", "F2_asthenie", "F3_depressif"),
  Alpha = c(
    round(cronbach(hdrs_J0[,hdrs_items])$alpha, 3),
    round(cronbach(hdrs_J0[,c("8_ralentissement", "12_symptomes_gastro", "14_symptomes_genitiaux", "16_perte_poids")])$alpha, 3),
    round(cronbach(hdrs_J0[,c("4_insomnie_debut", "13_symptomes_generaux")])$alpha, 3),
    round(cronbach(hdrs_J0[,c("1_humeur_depressive", "2_sentim_culpabilite", "3_suicide", "5_insomnie_milieu", "6_insomnie_matin", "7_travail_activite", "9_agitation", "10_anxiete_psychique", "11_anxiete_somatique", "15_hypochondrie", "17_prise_conscience")])$alpha, 3)
  ),
  CI_lower = c(
    round(boot.ci(boot_alpha_global, type = "bca")$bca[4], 3),
    round(boot.ci(boot_alpha_f1, type = "bca")$bca[4], 3),
    round(boot.ci(boot_alpha_f2, type = "bca")$bca[4], 3),
    round(boot.ci(boot_alpha_f3, type = "bca")$bca[4], 3)
  ),
  CI_upper = c(
    round(boot.ci(boot_alpha_global, type = "bca")$bca[5], 3),
    round(boot.ci(boot_alpha_f1, type = "bca")$bca[5], 3),
    round(boot.ci(boot_alpha_f2, type = "bca")$bca[5], 3),
    round(boot.ci(boot_alpha_f3, type = "bca")$bca[5], 3)
  )
)
knitr::kable(
  alpha_df,
  caption = "Alpha de Cronbach et intervalles de confiance à 95% pour l'échelle de Hamilton à J0 et ses sous-échelles",
  booktabs = TRUE,
  align = "lccc"
)

#création fichier large avec hdrs_J0 et scl90_dim à J0
scl90_J0 <- subset(scl90_dim, VISIT == "J0")
scl90_J0 <- scl90_J0[,c("NUMERO", "VISIT", dimensions)]

```

```

scl90_J0 <- scl90_J0[order(scl90_J0$NUMERO), ]
hdrs_J0 <- hdrs_J0[order(hdrs_J0$NUMERO), ]
hdrs_scl90_J0 <- merge(hdrs_J0, scl90_J0, by = c("NUMERO", "VISIT"), all.x = TRUE)

# Matrice de corrélation complète
corr_validite_J0 <- cor(
  hdrs_scl90_J0[, c("score", dimensions)],
  use = "pairwise.complete.obs"
)

# Corrplot
corrplot(
  corr_validite_J0["score", , drop = FALSE],
  method = "color",
  type = "full",
  tl.col = "black",
  addCoef.col = "white",
  number.cex = 0.5,
  tl.cex = 0.4,
  tl.srt = 30,
  col = viridis::plasma(100)
)

corr_validite_sousscores_J0 <- cor(hdrs_scl90_J0[,c("f1_somatique", "f2_asthenie", "f3_depressif",
  dimensions)], use = "pairwise.complete.obs")

# Corrplot
corrplot(
  corr_validite_sousscores_J0[c("f1_somatique", "f2_asthenie", "f3_depressif"), , drop = FALSE],
  method = "color",
  type = "full",
  tl.col = "black",
  addCoef.col = "white",
  number.cex = 0.5,
  tl.cex = 0.4,
  tl.srt = 30,
  col = viridis::plasma(100)
)

n_J56 <- nrow(subset(hdrs_groupe, VISIT == "J56"))
n_J56
n_J0 <- nrow(subset(hdrs_groupe, VISIT == "J0"))
n_J0
pct_present_J56 <- (n_J56 / n_J0) * 100
pct_present_J56

hdrs_J56 <- subset(hdrs_groupe, VISIT == "J56")
# Vraie palette Nord (package nord)
# On prend une palette qualitative (aurora) et on l'étend à 5 couleurs
cols_nord <- nord::nord("aurora", 5)
# Première série de graphiques (9 au maximum)
par(mfrow = c(3, 3), mar = c(2, 2, 2, 1))
items1 <- hdrs_items[1:min(9, length(hdrs_items))]
for (item in items1) {
  val <- na.omit(hdrs_J56[[item]])
  if (length(val) > 0) {
    m <- max(val)
    counts <- table(factor(val, levels = 0:m))
    barplot(counts,
      main = item,
      xlab = "Score",
      ylab = "Fréquence",
      col = cols_nord[1:(m + 1)],
      border = "white")
  } else {
    plot.new()
    title(main = paste(item, "(pas de données)"))
  }
}
par(mfrow = c(1, 1))
# Deuxième série de graphiques (8 au maximum, de 10 à 17)
if (length(hdrs_items) > 9) {

```

```

par(mfrow = c(3, 3), mar = c(2, 2, 2, 1))
items2 <- hdrs_items[10:min(17, length(hdrs_items))]
for (item in items2) {
  val <- na.omit(hdrs_J56[[item]])
  if (length(val) > 0) {
    m <- max(val)
    counts <- table(factor(val, levels = 0:m))
    barplot(counts,
            main = item,
            xlab = "Score",
            ylab = "Fréquence",
            col = cols_nord[1:(m + 1)],
            border = "white")
  } else {
    plot.new()
    title(main = paste(item, "(pas de données)"))
  }
}
par(mfrow = c(1, 1))
}

hdrs_J56_matrix <- hdrs_J56[, hdrs_items]
corr_matrix_J56 <- cor(hdrs_J56_matrix, use = "pairwise.complete.obs")

corrplot(corr_matrix_J56,
         method = "color",
         type = "upper",
         tl.col = "black",
         addCoef.col = "white",
         number.cex = 0.35,
         tl.cex = 0.6,
         tl.srt = 45,
         col = viridis::plasma(100)
        )

#renommer les noms des variables dans un nouveau df copié pour éviter la superposition
hdrs_J56_PCA <- hdrs_J56[,c(hdrs_items)]
colnames(hdrs_J56_PCA) <- c("1","2","3","4","5","6",
                           "7","8","9","10","11","12",
                           "13","14","15","16","17")

mdspca(hdrs_J56_PCA)

head(hdrs_J56[,c(hdrs_items)])
scree.plot(hdrs_J56[,c(hdrs_items)], simu=20, use = "P")

af_J56 <- factanal(
  na.omit(hdrs_J56[hdrs_items]),
  factors = 1,
  rotation = "varimax"
)
# Extraction propre des loadings
loadings_df_J56 <- as.data.frame(unclass(af_J56$loadings))
# Création explicite de la colonne Variable à partir des rownames
loadings_df_J56$Variable <- rownames(loadings_df_J56)
# Suppression des rownames pour éviter toute ambiguïté
rownames(loadings_df_J56) <- NULL
# Réorganisation : Variable en première colonne
loadings_df_J56 <- loadings_df_J56[, c("Variable", colnames(loadings_df_J56)[1])]
# Arrondi
loadings_df_J56[, -1] <- round(loadings_df_J56[, -1], 3)
# Affichage LaTeX
knitr::kable(
  loadings_df_J56,
  caption = "Contribution du facteur unique à la variance de la réponse à chaque item du score de Hamilton évalué à J56 (analyse factorielle avec rotation varimax à 1 facteur)",
  booktabs = TRUE,
  align = "lccc"
)

cronbach(hdrs_J56[,hdrs_items])

#estimation des IC par bootstrap
set.seed(123)

```

```

boot_alpha_J56 <- boot(hdrs_J56[,hdrs_items], alpha_bootstrap, R = 1000)
boot.ci(boot_alpha_J56, type = "bca")

#représentation des alpha de Cronbach avec IC en tableau
alpha_J56_df <- data.frame(
  Scale = c("Global"),
  Alpha = c(
    round(cronbach(hdrs_J56[,hdrs_items])$alpha, 3)
  ),
  CI_lower = c(
    round(boot.ci(boot_alpha_J56, type = "bca")$bca[4], 2)
  ),
  CI_upper = c(
    round(boot.ci(boot_alpha_J56, type = "bca")$bca[5], 2)
  )
)
knitr::kable(
  alpha_J56_df,
  caption = "Alpha de Cronbach et intervalle de confiance à 95% pour l'échelle de Hamilton à J56",
  booktabs = TRUE,
  align = "lccc"
)

#création fichier large avec hdrs_J56 et scl90_dim à J56
scl90_J56 <- subset(scl90_dim, VISIT == "J56")
scl90_J56 <- scl90_J56[,c("NUMERO", "VISIT", dimensions)]
scl90_J56 <- scl90_J56[order(scl90_J56$NUMERO), ]
hdrs_J56 <- hdrs_J56[order(hdrs_J56$NUMERO), ]
hdrs_scl90_J56 <- merge(hdrs_J56, scl90_J56, by = c("NUMERO", "VISIT"), all.x = TRUE)

# Matrice de corrélation complète
corr_validite_J56 <- cor(
  hdrs_scl90_J56[, c("score", dimensions)],
  use = "pairwise.complete.obs"
)

# Corrplot
corrplot(
  corr_validite_J56["score", , drop = FALSE],
  method = "color",
  type = "full",
  tl.col = "black",
  addCoef.col = "white",
  number.cex = 0.65,
  tl.cex = 0.4,
  tl.srt = 25,
  col = viridis::magma(100)
)

corr_scores_J56 <- corr_validite_J56["score", dimensions]
corr_max_J56 <- max(corr_scores_J56, na.rm = TRUE)
corr_max_dim_J56 <- names(which.max(corr_scores_J56))

df_fpca_J56 <- data.frame(
  score_total = hdrs_scl90_J56$score,
  hdrs_scl90_J56[, dimensions]
)

fpca(
  score_total ~ .,
  data = df_fpca_J56
)

library(dplyr)
hdrs_J0_grouped <- hdrs_J0 %>%
  group_by(GROUPE) %>%
  summarise(
    N = n(),
    Moyenne = round(mean(score, na.rm = TRUE), 1),
    "Écart_type" = round(sd(score, na.rm = TRUE), 2),
    Médiane = round(median(score, na.rm = TRUE), 2),
    Q1 = round(quantile(score, 0.25, na.rm = TRUE), 2),
    Q3 = round(quantile(score, 0.75, na.rm = TRUE), 2)
  )

```

```

)
knitr::kable(
  hdrs_J0_grouped,
  caption = "Scores_bruts_de_l'échelle_de_Hamilton_à_J0_par_groupe_de_traitement",
  booktabs = TRUE,
  align = "lcccccc"
)

par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
# Diagramme en bâtons des scores bruts par groupe (côté à côté)
counts <- table(hdrs_J0$score, hdrs_J0$GROUPE)
barplot(t(counts),
  main = "",
  xlab = "Score_brut_Hamilton",
  ylab = "Fréquence",
  col = c("#bf616a", "#88c0d0"),
  legend.text = c("Groupe_0", "Groupe_1"),
  args.legend = list(x = "topright"),
  beside = TRUE) # Boxplot
boxplot(score ~ GROUPE,
  data = hdrs_J0,
  main = "",
  xlab = "Groupe_de_traitement",
  ylab = "Score_brut_Hamilton",
  col = c("#bf616a", "#88c0d0"),
  names = c("Groupe_0", "Groupe_1")
)
par(mfrow = c(1, 1))
t_test_J0 <- hdrs_J0 %>%
  tbl_summary(
    include = score,
    by = GROUPE,
    statistic = all_continuous() ~ "{mean}_{sd}",
    digits = all_continuous() ~ 2
  ) %>%
  add_p(
    test = all_continuous() ~ "t.test",
    test.args = all_continuous() ~ list(var.equal = TRUE)
  )
t_test_J0

visits <- c("J0", "J4", "J7", "J14", "J21", "J28", "J42", "J56")
tableau_patients_jours <- table(
  hdrs_groupe$GROUPE,
  factor(hdrs_groupe$VISIT, levels = visits)
)
knitr::kable(
  tableau_patients_jours,
  caption = "Nombre_de_patients_par_groupe_de_traitement_et_visite",
  booktabs = TRUE,
  align = c("lcccccccc")
)

par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
# Groupe 0
hdrs_groupe0 <- subset(hdrs_groupe, GROUPE == 0)
boxplot(score ~ VISIT,
  data = hdrs_groupe0,
  main = "Groupe_0",
  xlab = "",
  ylab = "Score_brut_Hamilton",
  col = "#bf616a"
)
# Groupe 1
hdrs_groupe1 <- subset(hdrs_groupe, GROUPE == 1)
boxplot(score ~ VISIT,
  data = hdrs_groupe1,
  main = "Groupe_1",
  xlab = "",
  ylab = "",

```

```

    col = "#88c0d0"
  )
  par(mfrow = c(1, 1))

  # utilisation de ggplot2 pour les diagrammes en spaghetti avec ggparcoord
  library(ggplot2)
  library(GGally)
  palette_nord <- c("#bf616a", "#88c0d0")

  hdrs_wide <- reshape2::dcast(
    hdrs_groupe,
    NUMERO + GROUPE ~ VISIT,
    value.var = "score"
  )
  hdrs_wide$GROUPE <- factor(hdrs_wide$GROUPE)
  ggparcoord(
    data = hdrs_wide,
    columns = 3:10,
    groupColumn = "GROUPE",
    scale = "globalminmax",
    missing = "exclude"
  ) +
    theme_minimal() +
    labs(
      title = "",
      x = "",
      y = "Score_brut_Hamilton",
      color = "Groupe"
    ) +
    scale_color_manual(
      values = palette_nord,
      labels = c("Groupe_0", "Groupe_1")
    )
  )

  hdrs_J56 <- subset(hdrs_groupe, VISIT %in% c("J56"))
  boxplot(score ~ GROUPE,
    data = hdrs_J56,
    main = "",
    xlab = "",
    ylab = "Score_brut_Hamilton",
    col = c("#bf616a", "#88c0d0"),
    names = c("Groupe_0", "Groupe_1")
  )

  #on reprend hdrs_wide créé précédemment (hdrs au format large, qui contient les scores à chaque
  #visite en colonnes)
  hdrs_wide

  hdrs_wide_locf <- hdrs_wide
  #fonction na.locf de zoo pour imputer les valeurs manquantes par la dernière valeur observée
  hdrs_wide_locf[, visits] <- t(zoo::na.locf(t(as.matrix(hdrs_wide[, visits])), na.rm = FALSE))
  #vérification de l'imputation (format wide)
  hdrs_wide_locf

  # ajout d'une variable "difference" qui correspond à la différence entre le score à J56 et le
  #score à J0 (après imputation LOCF)
  hdrs_wide_locf$difference <- hdrs_wide_locf$J56 - hdrs_wide_locf$J0

  # retour au format long après imputation LOCF
  head(hdrs_wide_locf)
  hdrs_groupe_locf <- reshape(
    data = hdrs_wide_locf,
    idvar = c("NUMERO", "GROUPE"),
    varying = visits,
    times = visits,
    v.names = "score",
    timevar = "VISIT",
    direction = "long"
  )
  head(hdrs_groupe_locf)

  # garder aussi le score observé (avant imputation) dans le même df
  hdrs_groupe_obs <- reshape(

```

```

    data = hdrs_wide,
    idvar = c("NUMERO", "GROUPE"),
    varying = visits,
    times = visits,
    v.names = "score_obs",
    timevar = "VISIT",
    direction = "long"
)

# Effectifs après LOCF (toutes les visites existent maintenant)
tableau_patients_jours_locf <- table(hdrs_groupe_locf$GROUPE, hdrs_groupe_locf$VISIT)
tableau_patients_jours_locf

knitr::kable(
  tableau_patients_jours_locf,
  caption = "Nombre de patients par groupe de traitement et visite après imputation LOCF",
  booktabs = TRUE,
  align = c("lccccccc")
)

par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
counts <- table(hdrs_wide_locf$difference)
barplot(counts,
  main = "",
  xlab = "Différence de score (J56-J0)",
  ylab = "Fréquence",
  col = "#81a1c1",
  border = "white"
)

qqnorm(hdrs_wide_locf$difference,
  main = "",
  col = "#81a1c1"
)

qqline(hdrs_wide_locf$difference, col = "red")
par(mfrow = c(1, 1))

t_test_diff_locf <- hdrs_wide_locf %>%
  tbl_summary(
    include = difference,
    by = GROUPE,
    statistic = all_continuous() ~ "{mean}_{sd}",
    digits = all_continuous() ~ 2
  ) %>%
  add_p(
    test = all_continuous() ~ "t.test",
    test.args = all_continuous() ~ list(var.equal = TRUE)
  )
t_test_diff_locf

# ordonner les niveaux de VISIT pour avoir J0 en référence et J56 en dernier
# ne pas faire sur les données imputées (données AVANT LOCF)
hdrs_groupe$VISIT <- factor(as.character(hdrs_groupe$VISIT),
  levels = c("J0", "J4", "J7", "J14", "J21", "J28", "J42", "J56"))

modele_mixte_sans_interaction <- lmer(
  score ~ VISIT + GROUPE + (1 | NUMERO),
  data = hdrs_groupe
)

summary(modele_mixte_sans_interaction)

# Extraction des coefficients, IC et p-values
coef_summary <- summary(modele_mixte_sans_interaction)$coefficients
conf_int <- confint(modele_mixte_sans_interaction, level = 0.95, parm = "beta_")
p_values <- coef_summary[, "Pr(>|t|)"]
p_values_fmt <- ifelse(
  p_values < 0.001,
  "<0.001",
  ifelse(
    p_values < 0.01,
    "<0.01",
    ifelse(p_values < 0.05, "<0.05", sprintf("%.4f", p_values))
  )
)

```

```

)
# Création du tableau récapitulatif
modele_mixte_sans_interaction_df <- data.frame(
  Coefficient = rownames(coef_summary),
  Estimate = round(coef_summary[, "Estimate"], 3),
  Std_Error = round(coef_summary[, "Std. Error"], 3),
  t_value = round(coef_summary[, "t_value"], 3),
  CI_lower = round(conf_int[, 1], 3),
  CI_upper = round(conf_int[, 2], 3),
  p_value = p_values_fmt
)
# Affichage LaTeX
knitr::kable(
  modele_mixte_sans_interaction_df,
  caption = "Estimation des coefficients du modèle mixte sans interaction entre VISIT (facteur) et GROUPE",
  booktabs = TRUE,
  align = "lcccccc",
  row.names = FALSE
)

# convertir VISIT en variable numérique (jours)
hdrs_groupe$VISIT_num <- as.numeric(sub("J", "", as.character(hdrs_groupe$VISIT)))
modele_mixte_sans_interaction_continu <- lmer(
  score ~ VISIT_num + GROUPE + (1 | NUMERO),
  data = hdrs_groupe
)
summary(modele_mixte_sans_interaction_continu)

# Extraction des coefficients, IC et p-values
coef_summary_continu <- summary(modele_mixte_sans_interaction_continu)$coefficients
conf_int_continu <- confint(modele_mixte_sans_interaction_continu, level = 0.95, parm = "beta_")
p_values_continu <- coef_summary_continu[, "Pr(>|t|)"]
p_values_continu_fmt <- ifelse(
  p_values_continu < 0.001,
  "<0.001",
  ifelse(
    p_values_continu < 0.01,
    "<0.01",
    ifelse(p_values_continu < 0.05, "<0.05", sprintf("%.4f", p_values_continu))
  )
)

# Création du tableau récapitulatif
modele_mixte_sans_interaction_continu_df <- data.frame(
  Coefficient = rownames(coef_summary_continu),
  Estimate = round(coef_summary_continu[, "Estimate"], 3),
  Std_Error = round(coef_summary_continu[, "Std. Error"], 3),
  t_value = round(coef_summary_continu[, "t_value"], 3),
  CI_lower = round(conf_int_continu[, 1], 3),
  CI_upper = round(conf_int_continu[, 2], 3),
  p_value = p_values_continu_fmt
)
# Affichage LaTeX
knitr::kable(
  modele_mixte_sans_interaction_continu_df,
  caption = "Estimation des coefficients du modèle mixte sans interaction entre VISIT (continue) et GROUPE",
  booktabs = TRUE,
  align = "lcccccc",
  row.names = FALSE
)

modele_mixte_avec_interaction <- lmer(
  score ~ VISIT * GROUPE + (1 | NUMERO),
  data = hdrs_groupe
)
summary(modele_mixte_avec_interaction)

# Extraction des coefficients, IC et p-values
coef_summary <- summary(modele_mixte_avec_interaction)$coefficients
p_values <- coef_summary[, "Pr(>|t|)"]
p_values_fmt <- ifelse(
  p_values < 0.001,

```



```

    "<0.001",
    ifelse(
      p_values < 0.01,
      "<0.01",
      ifelse(p_values < 0.05, "<0.05", sprintf("%.4f", p_values))
    )
  )
}
# Création du tableau récapitulatif
modele_mixte_interaction_df <- data.frame(
  Coefficient = rownames(coef_summary),
  Estimate = round(coef_summary[, "Estimate"], 3),
  Std_Error = round(coef_summary[, "Std. Error"], 3),
  t_value = round(coef_summary[, "t_value"], 3),
  p_value = p_values_fmt
)
# Affichage LaTeX
knitr::kable(
  modele_mixte_interaction_df,
  caption = "Estimation des coefficients du modèle mixte avec interaction entre VISIT (facteur) et GROUPE",
  booktabs = TRUE,
  align = "lcccc",
  row.names = FALSE
)

modele_mixte_avec_interaction_continu <- lmer(
  score ~ VISIT_num * GROUPE + (1 | NUMERO),
  data = hdrs_groupe
)
summary(modele_mixte_avec_interaction_continu)

# Extraction des coefficients, IC et p-values
coef_summary_continu <- summary(modele_mixte_avec_interaction_continu)$coefficients
p_values_continu <- coef_summary_continu[, "Pr(>|t|)"]
p_values_continu_fmt <- ifelse(
  p_values_continu < 0.001,
  "<0.001",
  ifelse(
    p_values_continu < 0.01,
    "<0.01",
    ifelse(p_values_continu < 0.05, "<0.05", sprintf("%.4f", p_values_continu))
  )
)
# Création du tableau récapitulatif
modele_mixte_interaction_continu_df <- data.frame(
  Coefficient = rownames(coef_summary_continu),
  Estimate = round(coef_summary_continu[, "Estimate"], 3),
  Std_Error = round(coef_summary_continu[, "Std. Error"], 3),
  t_value = round(coef_summary_continu[, "t_value"], 3),
  p_value = p_values_continu_fmt
)
# Affichage LaTeX
knitr::kable(
  modele_mixte_interaction_continu_df,
  caption = "Estimation des coefficients du modèle mixte avec interaction entre VISIT (continue) et GROUPE",
  booktabs = TRUE,
  align = "lcccc",
  row.names = FALSE
)

test_interaction <- drop1(
  modele_mixte_avec_interaction_continu,
  test = "F"
)

num_df <- if ("NumDF" %in% colnames(test_interaction)) {
  test_interaction$NumDF
} else if ("Df" %in% colnames(test_interaction)) {
  test_interaction$Df
} else {
  rep(NA_integer_, nrow(test_interaction))
}

```

```

den_df <- if ("DenDF" %in% colnames(test_interaction)) {
  test_interaction$DenDF
} else {
  rep(NA_integer_, nrow(test_interaction))
}
f_value <- if ("F_value" %in% colnames(test_interaction)) {
  test_interaction$`F value`
} else if ("F" %in% colnames(test_interaction)) {
  test_interaction$F
} else {
  rep(NA_real_, nrow(test_interaction))
}
test_interaction_df <- data.frame(
  Term = rownames(test_interaction),
  NumDF = num_df,
  DenDF = den_df,
  F_value = round(f_value, 3),
  p_value = ifelse(
    test_interaction$`Pr(>F)` < 0.001,
    "<0.001",
    ifelse(
      test_interaction$`Pr(>F)` < 0.01,
      "<0.01",
      ifelse(test_interaction$`Pr(>F)` < 0.05, "<0.05", sprintf("%.4f", test_interaction$`Pr
        (>F)`))
    )
  )
)
knitr::kable(
  test_interaction_df,
  caption = "Test_F_de_l'effet_d'interaction_entre_VISIT_(continue)_et_GROUPE",
  booktabs = TRUE,
  align = "lcccc",
  row.names = FALSE
)

par(mfrow = c(1, 2))
residus <- resid(modele_mixte_avec_interaction_continu)
hist(residus,
  main = "",
  xlab = "Résidus",
  col = "#81a1c1",
  border = "white",
  breaks = 100,
  freq = FALSE
)
qqnorm(residus,
  main = "",
  col = "#81a1c1"
)
qqline(residus, col = "red")
par(mfrow = c(1, 1))

# on repart de hdrs_wide (scores HDRS par visite)
hdrs_wide_reponse <- hdrs_wide
hdrs_wide_reponse$score_J0 <- hdrs_wide_reponse$J0

# extraire la dernière visite observée (non manquante) par patient
visits_matrix <- as.matrix(hdrs_wide_reponse[, visits])
valid <- rowSums(!is.na(visits_matrix)) > 0

# index de la dernière valeur non manquante sur la ligne
last_idx <- max.col(!is.na(visits_matrix), ties.method = "last")
last_idx[!valid] <- NA

# stocker la dernière visite et le dernier score observé
hdrs_wide_reponse$derniere_visite <- NA
hdrs_wide_reponse$score_dernier <- NA
hdrs_wide_reponse$derniere_visite[valid] <- visits[last_idx[valid]]
hdrs_wide_reponse$score_dernier[valid] <- visits_matrix[cbind(which(valid), last_idx[valid])]

# critère binaire de réponse: chute >= 50% par rapport à J0
hdrs_wide_reponse$reponse <- ifelse(

```

```

    !is.na(hdrs_wide_reponse$score_dernier) &
    hdrs_wide_reponse$score_dernier <= 0.5 * hdrs_wide_reponse$score_J0,
    1,
    0
)

# fusionner la réponse avec les données au format long
hdrs_groupe_reponse <- merge(
  hdrs_groupe,
  hdrs_wide_reponse[, c("NUMERO", "GROUPE", "score_J0", "derniere_visite", "score_dernier", "
    reponse")],
  by = c("NUMERO", "GROUPE")
)

# verification rapide du resultat (aperçu)
head(hdrs_groupe_reponse)

# verifier que hdrs_groupe et hdrs_groupe_reponse ne diffèrent pas hors colonnes ajoutées
cols_added <- c("score_J0", "derniere_visite", "score_dernier", "reponse")
common_cols <- setdiff(names(hdrs_groupe_reponse), cols_added)
hdrs_groupe_sorted <- hdrs_groupe[order(hdrs_groupe$NUMERO, hdrs_groupe$GROUPE), common_cols]
hdrs_groupe_reponse_base <- hdrs_groupe_reponse[order(hdrs_groupe_reponse$NUMERO, hdrs_groupe_
  reponse$GROUPE), common_cols]
coerce_like <- function(x, template) {
  if (is.factor(template)) return(factor(x, levels = levels(template)))
  if (is.integer(template)) return(as.integer(x))
  if (is.numeric(template)) return(as.numeric(x))
  if (is.character(template)) return(as.character(x))
  if (is.logical(template)) return(as.logical(x))
  x
}
for (col in common_cols) {
  hdrs_groupe_reponse_base[[col]] <- coerce_like(hdrs_groupe_reponse_base[[col]], hdrs_groupe_
    sorted[[col]])
}
all.equal(hdrs_groupe_sorted, hdrs_groupe_reponse_base, check.attributes = FALSE)

# prop.table: chute < 50% a la derniere visite (dans hdrs_groupe via jointure)
tmp_resp <- hdrs_wide_reponse[, c("NUMERO", "GROUPE", "score_J0", "score_dernier")]
tmp_resp$chute_lt_50 <- with(tmp_resp, !is.na(score_dernier) & score_dernier > 0.5 * score_J0)
hdrs_groupe_check <- merge(hdrs_groupe, tmp_resp[, c("NUMERO", "GROUPE", "chute_lt_50")], by = c(
  "NUMERO", "GROUPE"), all.x = TRUE)
prop.table(table(hdrs_groupe_check$chute_lt_50, useNA = "ifany"))

# prop.table: reponse binaire dans hdrs_groupe_reponse
prop.table(table(hdrs_groupe_reponse$reponse, useNA = "ifany"))

# convertir les visites en jours
visit_days <- c(J0 = 0, J4 = 4, J7 = 7, J14 = 14, J21 = 21, J28 = 28, J42 = 42, J56 = 56)
# calculer temps_suivi et evenement (réponse)
scores_mat <- as.matrix(hdrs_wide_reponse[, visits])
follow_mat <- scores_mat[, -1, drop = FALSE]
score_j0 <- hdrs_wide_reponse$score_J0

# position de la première visite manquante (censure à la visite précédente)
na_mat <- is.na(follow_mat)
na_any <- rowSums(na_mat) > 0
na_pos <- max.col(na_mat, ties.method = "first")
na_pos[!na_any] <- NA

# position de la première réponse (chute >= 50% par rapport à J0)
resp_mat <- sweep(follow_mat, 1, 0.5 * score_j0, "<=")
resp_mat[is.na(resp_mat)] <- FALSE
resp_any <- rowSums(resp_mat) > 0
resp_pos <- max.col(resp_mat, ties.method = "first")
resp_pos[!resp_any] <- NA

# par défaut : censuré à J56
temps_suivi <- rep(visit_days["J56"], nrow(follow_mat))
evenement <- rep(0, nrow(follow_mat))

# réponse avant la première NA
event_before_censor <- !is.na(resp_pos) & (is.na(na_pos) | resp_pos < na_pos)

```

```

temps_suivi[event_before_censor] <- visit_days[visits[-1][resp_pos[event_before_censor]]]
evenement[event_before_censor] <- 1

# censure avant réponse
censor_before_event <- !is.na(na_pos) & (is.na(resp_pos) | na_pos < resp_pos)
temps_suivi[censor_before_event] <- visit_days[visits[na_pos[censor_before_event]]]

hdrs_wide_reponse$temps_suivi <- temps_suivi
hdrs_wide_reponse$evenement <- evenement
# vérifier les temps de suivi
head(hdrs_wide_reponse[, c("NUMERO", "temps_suivi", "evenement", "reponse")])
# distribution des événements par temps (doit montrer des réponses avant J56)
table(hdrs_wide_reponse$evenement, hdrs_wide_reponse$temps_suivi, useNA = "ifany")

tableau_temps_suivi <- table(
  hdrs_wide_reponse$GROUPE,
  hdrs_wide_reponse$temps_suivi
)
knitr::kable(
  tableau_temps_suivi,
  caption = "Temps de suivi (en jours) par groupe de traitement",
  booktabs = TRUE,
  align = c("lccccccc")
)

# créer l'objet Survival
surv_obj_reponse <- Surv(time = hdrs_wide_reponse$temps_suivi, event = hdrs_wide_reponse$
  evenement)
# ajuster le modèle de survie de Kaplan-Meier par groupe
km_fit_reponse <- survfit(surv_obj_reponse ~ GROUPE, data = hdrs_wide_reponse)
# test de log-rank
logrank_test_reponse <- survdiff(surv_obj_reponse ~ GROUPE, data = hdrs_wide_reponse)
logrank_test_reponse

# tracer les courbes de survie
ggsurvplot(km_fit_reponse,
  data = hdrs_wide_reponse,
  pval = TRUE,
  conf.int = TRUE,
  risk.table = TRUE,
  risk.table.col = "strata",
  xlim = c(0, 56),
  break.time.by = 7,
  surv.scale = "percent",
  legend.labs = c("Groupe_0", "Groupe_1"),
  palette = c("#bf616a", "#88c0d0"),
  xlab = "Temps(jours)",
  ylab = "Probabilité de survie",
)

# ajuster le modèle de Cox
cox_model_reponse <- coxph(surv_obj_reponse ~ GROUPE, data = hdrs_wide_reponse)

# Extraction des coefficients, HR, IC et p-values
coef_summary_cox <- summary(cox_model_reponse)$coefficients
conf_int_cox <- confint(cox_model_reponse, level = 0.95)
# Création du tableau récapitulatif
cox_model_reponse_df <- data.frame(
  Coefficient = rownames(coef_summary_cox),
  Estimate = round(coef_summary_cox[, "coef"], 3),
  HR = round(exp(coef_summary_cox[, "coef"]), 3),
  CI_lower = round(exp(conf_int_cox[, 1]), 3),
  CI_upper = round(exp(conf_int_cox[, 2]), 3),
  p_value = round(coef_summary_cox[, "Pr(>|z|)"], 4)
)

# Affichage LaTeX
knitr::kable(
  cox_model_reponse_df,
  caption = "Estimation des coefficients du modèle de Cox pour le critère binaire de réponse au
    traitement",
  booktabs = TRUE,
  align = "lcccc",
  row.names = FALSE
)

```

```

)

plot(cox.zph(cox_model_reponse), main = "Risques proportionnels - Modèle de Cox binaire")

#insérer le code à la fin du fichier

# lire le fichier code genere
code <- readLines("all_chunks_code.R", warn = FALSE)
code_wrapped <- code

if (knitr::is_latex_output()) {
  code_safe <- gsub(
    "\\\\(begin|end)\\{lstlisting\\}",
    "\\\textbackslash\\{lstlisting}",
    code_wrapped
  )
  cat("\\textbackslashbegin{lstlisting}\n")
  cat(code_safe, sep = "\n")
  cat("\n\\textbackslashend{lstlisting}\n")
} else {
  cat("```\r\n")
  cat(code_wrapped, sep = "\n")
  cat("\n```\n")
}

```