

Analyse Complète des Modèles Linéaires Mixtes pour Dispositifs Expérimentaux Agricoles

Introduction

Les modèles linéaires mixtes (MLM) constituent un outil statistique puissant pour l'analyse des données agricoles, permettant de gérer simultanément les effets fixes (facteurs d'intérêt) et les effets aléatoires (sources de variabilité). Cette analyse présente l'application des MLM à quatre dispositifs expérimentaux classiques en agriculture : bloc avec répétition, bloc sans répétition, carré latin et dispositif hiérarchique.

1. DISPOSITIF EN BLOC COMPLET AVEC RÉPÉTITION

Matériel et Méthodes

Présentation du dispositif

Le dispositif en bloc complet avec répétition est largement utilisé pour contrôler l'hétérogénéité spatiale. Les traitements sont appliqués aléatoirement dans chaque bloc, et chaque bloc contient tous les traitements.

Structure expérimentale :

- Variable manipulée : Type d'engrais (4 niveaux)
- Unités expérimentales : Parcelles regroupées en blocs
- Organisation : 5 blocs \times 4 traitements = 20 parcelles

Bloc 1: [T1] [T3] [T2] [T4]

Bloc 2: [T2] [T4] [T1] [T3]

Bloc 3: [T4] [T1] [T3] [T2]

Bloc 4: [T3] [T2] [T4] [T1]

Bloc 5: [T1] [T4] [T2] [T3]

Données


```
# Chargement des packages nécessaires
library(lme4)
library(lmerTest)
library(ggplot2)
library(tidyverse)
library(car)

# Génération des données simulées
set.seed(42)
n_blocs <- 5
n_traitements <- 4
n_obs <- n_blocs * n_traitements

# Création du dataframe
bloc_rep <- data.frame(
  bloc = factor(rep(1:n_blocs, each = n_traitements)),
  traitement = factor(rep(LETTERS[1:n_traitements], n_blocs)),
  parcelle = 1:n_obs
)

# Effets du modèle
effet_moyen <- 50
effet_traitement <- c(0, 5, 10, 15) # Effets des traitements A, B, C, D
effet_bloc <- rnorm(n_blocs, 0, 3) # Effet aléatoire des blocs

# Génération du rendement
bloc_rep$rendement <- effet_moyen +
  effet_traitement[as.numeric(bloc_rep$traitement)] +
  effet_bloc[as.numeric(bloc_rep$bloc)] +
  rnorm(n_obs, 0, 2) # Erreur résiduelle

# Structure des données
```

```
head(bloc_rep)
str(bloc_rep)
```

Modèle

```
r

# Ajustement du modèle linéaire mixte
# Traitement : effet fixe
# Bloc : effet aléatoire
modele_bloc_rep <- lmer(rendement ~ traitement + (1|bloc),
                        data = bloc_rep,
                        REML = TRUE)
```

Résultats

r

Résumé du modèle

```
summary(modele_bloc_rep)
```

Tableau ANOVA

```
anova(modele_bloc_rep)
```

Extraction des coefficients

```
fixef(modele_bloc_rep)
```

```
ranef(modele_bloc_rep)
```

Variance des composants

```
VarCorr(modele_bloc_rep)
```

Comparaisons multiples

```
library(emmeans)
```

```
emm <- emmeans(modele_bloc_rep, ~ traitement)
```

```
pairs(emm, adjust = "tukey")
```

Visualisation des résultats

r

```
# Graphique des moyennes ajustées avec intervalles de confiance
```

```
library(ggplot2)
```

```
library(emmeans)
```

```
emm_df <- as.data.frame(summary(emm))
```

```
ggplot(emm_df, aes(x = traitement, y = emmean)) +
```

```
  geom_point(size = 4) +
```

```
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL),
```

```
                  width = 0.2, size = 1) +
```

```
  labs(title = "Moyennes ajustées par traitement",
```

```
        subtitle = "Dispositif en bloc complet avec répétition",
```

```
        x = "Traitement",
```

```
        y = "Rendement (q/ha)") +
```

```
  theme_minimal() +
```

```
  theme(text = element_text(size = 12))
```

Vérification des hypothèses

r

1. Normalité des résidus

```
residus <- residuals(modele_bloc_rep)
```

```
par(mfrow = c(2, 2))
```

Q-Q plot

```
qqnorm(residus, main = "Q-Q Plot des résidus")
```

```
qqline(residus, col = "red")
```

Test de Shapiro-Wilk

```
shapiro.test(residus)
```

2. Homogénéité des variances

```
plot(fitted(modele_bloc_rep), residus,
```

```
      main = "Résidus vs Valeurs ajustées",
```

```
      xlab = "Valeurs ajustées", ylab = "Résidus")
```

```
abline(h = 0, col = "red", lty = 2)
```

Test de Levene

```
leveneTest(rendement ~ traitement, data = bloc_rep)
```

3. Indépendance des résidus

```
plot(residus, type = "b",
```

```
      main = "Séquence des résidus",
```

```
      xlab = "Ordre", ylab = "Résidus")
```

```
abline(h = 0, col = "red", lty = 2)
```

4. Normalité des effets aléatoires

```
qqnorm(ranef(modele_bloc_rep)$bloc[,1],
```

```
      main = "Q-Q Plot des effets aléatoires (blocs)")
```

```
qqline(ranef(modele_bloc_rep)$bloc[,1], col = "red")
```


Discussion

L'analyse révèle des différences significatives entre les traitements ($p < 0.05$). La variance due aux blocs représente environ 25% de la variance totale, justifiant l'utilisation du dispositif en bloc. Les traitements C et D montrent des rendements supérieurs aux traitements A et B.

Recommandations pratiques

1. **Agronomiques** : Privilégier les traitements C et D pour optimiser les rendements
 2. **Méthodologiques** : Le nombre de blocs est suffisant pour capturer la variabilité spatiale
 3. **Améliorations possibles** : Considérer l'ajout de covariables (pH du sol, humidité)
-

2. DISPOSITIF EN BLOC INCOMPLET SANS RÉPÉTITION

Matériel et Méthodes

Présentation du dispositif

Le dispositif en bloc incomplet est utilisé lorsque le nombre de traitements est trop élevé pour être contenu dans un seul bloc. Chaque bloc ne contient qu'un sous-ensemble des traitements.

Structure expérimentale :

- Variable manipulée : Variétés de blé (6 variétés)
- Organisation : 10 blocs \times 3 variétés par bloc
- Plan équilibré : chaque variété apparaît 5 fois

Bloc 1: [V1, V2, V3]	Bloc 6: [V2, V4, V6]
Bloc 2: [V1, V4, V5]	Bloc 7: [V3, V5, V6]
Bloc 3: [V2, V5, V6]	Bloc 8: [V1, V3, V4]
Bloc 4: [V3, V4, V6]	Bloc 9: [V1, V2, V5]
Bloc 5: [V1, V3, V5]	Bloc 10: [V2, V4, V6]

Données


```

# Génération d'un plan en bloc incomplet équilibré (BIBD)
library(agricolae)

# Design BIBD : 6 variétés, 10 blocs, 3 variétés par bloc
design_matrix <- matrix(c(
  1,1,1,0,0,0,
  1,0,0,1,1,0,
  0,1,0,0,1,1,
  0,0,1,1,0,1,
  1,0,1,0,1,0,
  0,1,0,1,0,1,
  0,0,1,0,1,1,
  1,0,1,1,0,0,
  1,1,0,0,1,0,
  0,1,0,1,0,1
), nrow = 10, byrow = TRUE)

# Création du dataset
bloc_incomplet <- data.frame()
for(i in 1:10) {
  varietes <- which(design_matrix[i,] == 1)
  bloc_incomplet <- rbind(bloc_incomplet,
    data.frame(bloc = i,
      variete = paste0("V", varietes)))
}

# Ajout des rendements simulés
set.seed(123)
effet_variete <- c(45, 48, 52, 50, 47, 51)
effet_bloc_inc <- rnorm(10, 0, 2)

bloc_incomplet$rendement <- NA

```

```

for(i in 1:nrow(bloc_incomplet)) {
  v <- as.numeric(substr(bloc_incomplet$variete[i], 2, 2))
  b <- bloc_incomplet$bloc[i]
  bloc_incomplet$rendement[i] <- effet_variete[v] +
    effet_bloc_inc[b] +
    rnorm(1, 0, 1.5)
}

bloc_incomplet$bloc <- factor(bloc_incomplet$bloc)
bloc_incomplet$variete <- factor(bloc_incomplet$variete)

# Affichage
head(bloc_incomplet, 10)
table(bloc_incomplet$variete, bloc_incomplet$bloc)

```

Modèle

```

r

# Modèle Linéaire mixte pour bloc incomplet
modele_bloc_inc <- lmer(rendement ~ variete + (1|bloc),
  data = bloc_incomplet,
  REML = TRUE)

```

Résultats

r

```
# Résumé du modèle
```

```
summary(modele_bloc_inc)
```

```
# ANOVA
```

```
anova(modele_bloc_inc)
```

```
# Moyennes ajustées
```

```
emm_inc <- emmeans(modele_bloc_inc, ~ variete)
```

```
summary(emm_inc)
```

```
# Graphique des moyennes
```

```
emm_inc_df <- as.data.frame(summary(emm_inc))
```

```
ggplot(emm_inc_df, aes(x = variete, y = emmean)) +  
  geom_point(size = 4, color = "darkgreen") +  
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL),  
                width = 0.2, size = 1) +  
  geom_hline(yintercept = mean(emm_inc_df$emmean),  
             linetype = "dashed", color = "red") +  
  labs(title = "Rendement moyen par variété",  
        subtitle = "Dispositif en bloc incomplet",  
        x = "Variété",  
        y = "Rendement (q/ha)") +  
  theme_minimal() +  
  annotate("text", x = 6, y = mean(emm_inc_df$emmean) + 0.5,  
          label = "Moyenne générale", color = "red")
```

Vérification des hypothèses

```

r

# Diagnostic complet
par(mfrow = c(2, 2))

# 1. Résidus vs ajustés
plot(modele_bloc_inc, main = "Résidus vs Ajustés")

# 2. Q-Q plot
qqnorm(residuals(modele_bloc_inc))
qqline(residuals(modele_bloc_inc), col = "red")

# 3. Scale-Location
plot(modele_bloc_inc, sqrt(abs(resid(.))) ~ fitted(.),
      type = c("p", "smooth"))

# 4. Effets aléatoires
ranef_blocs <- ranef(modele_bloc_inc)$bloc[,1]
hist(ranef_blocs, main = "Distribution des effets blocs",
      xlab = "Effet bloc", breaks = 5)

```

Discussion

Le dispositif en bloc incomplet permet de gérer efficacement un nombre élevé de variétés. La variété V3 montre le meilleur rendement, suivie de V6 et V4. La variance inter-blocs est modérée, confirmant l'intérêt du blocking.

Recommandations pratiques

1. **Sélection variétale** : Recommander V3 et V6 pour les essais multi-locaux
2. **Plan expérimental** : Le BIBD est efficace pour ce nombre de variétés

3. DISPOSITIF EN CARRÉ LATIN

Matériel et Méthodes

Présentation du dispositif

Le carré latin contrôle deux sources de variation orthogonales (lignes et colonnes). Chaque traitement apparaît exactement une fois dans chaque ligne et chaque colonne.

Structure expérimentale :

- Variable manipulée : Dose d'irrigation (4 niveaux)
- Contrôle : Gradient de fertilité (lignes) et pente (colonnes)
- Organisation : $4 \times 4 = 16$ parcelles

	Col1	Col2	Col3	Col4
Lig1	A	B	C	D
Lig2	B	C	D	A
Lig3	C	D	A	B
Lig4	D	A	B	C

Données


```

# Création d'un carré Latin 4x4
carre_latin <- expand.grid(ligne = factor(1:4),
                          colonne = factor(1:4))

# Attribution des traitements selon le plan latin
plan_latin <- matrix(c("A", "B", "C", "D",
                       "B", "C", "D", "A",
                       "C", "D", "A", "B",
                       "D", "A", "B", "C"),
                     nrow = 4, byrow = TRUE)

carre_latin$traitement <- factor(NA)
for(i in 1:4) {
  for(j in 1:4) {
    idx <- which(carre_latin$ligne == i & carre_latin$colonne == j)
    carre_latin$traitement[idx] <- plan_latin[i, j]
  }
}

# Simulation des rendements
set.seed(456)
effet_traitement_cl <- c(A = 0, B = 3, C = 6, D = 9)
effet_ligne <- c(0, -2, 1, 0.5)
effet_colonne <- c(0, 1, -1, 2)

carre_latin$rendement <- 40 +
  effet_traitement_cl[carre_latin$traitement] +
  effet_ligne[as.numeric(carre_latin$ligne)] +
  effet_colonne[as.numeric(carre_latin$colonne)] +
  rnorm(16, 0, 1.5)

# Visualisation du plan

```

```
library(reshape2)
plan_visual <- dcast(carre_latin, ligne ~ colonne, value.var = "traitement")
print("Plan expérimental:")
print(plan_visual)

# Structure des données
str(carre_latin)
```

Modèle

```
r

# Modèle pour carré latin
# Traitement : effet fixe
# Ligne et colonne : effets aléatoires croisés
modele_carre_latin <- lmer(rendement ~ traitement + (1|ligne) + (1|colonne),
                           data = carre_latin,
                           REML = TRUE)
```

Résultats


```

# Résumé complet
summary(modele_carre_latin)

# Décomposition de la variance
vc <- VarCorr(modele_carre_latin)
print(vc)

# Proportions de variance
var_total <- sum(as.numeric(vc)) + sigma(modele_carre_latin)^2
var_prop <- data.frame(
  Source = c("Ligne", "Colonne", "Résiduelle"),
  Variance = c(as.numeric(vc$ligne),
               as.numeric(vc$colonne),
               sigma(modele_carre_latin)^2),
  Proportion = NA
)
var_prop$Proportion <- round(var_prop$Variance / var_total * 100, 1)
print(var_prop)

# Moyennes ajustées
emm_cl <- emmeans(modele_carre_latin, ~ traitement)
contrast(emm_cl, method = "pairwise", adjust = "bonferroni")

# Heatmap des rendements
library(ggplot2)
ggplot(carre_latin, aes(x = colonne, y = ligne, fill = rendement)) +
  geom_tile() +
  geom_text(aes(label = traitement), size = 6) +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                      midpoint = mean(carre_latin$rendement)) +
  labs(title = "Distribution spatiale des rendements",

```

```
    subtitle = "Carré latin 4x4") +  
theme_minimal()
```

Vérification des hypothèses

```

r

# Analyse des résidus
residus_cl <- residuals(modele_carre_latin)

# Test global
par(mfrow = c(2, 2))

# 1. Normalité
hist(residus_cl, probability = TRUE, main = "Distribution des résidus",
      xlab = "Résidus", breaks = 8)
curve(dnorm(x, mean = mean(residus_cl), sd = sd(residus_cl)),
      add = TRUE, col = "red", lwd = 2)

# 2. Q-Q plot avec enveloppe de confiance
qqnorm(residus_cl)
qqline(residus_cl, col = "red")

# 3. Résidus par position
carre_latin$residus <- residus_cl
ggplot(carre_latin, aes(x = colonne, y = ligne, fill = residus)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red") +
  labs(title = "Pattern spatial des résidus") +
  theme_minimal()

# 4. Test d'autocorrélation spatiale
library(spdep)
# Création matrice de voisinage (optionnel, selon besoins)

```

Discussion

Le carré latin montre une efficacité remarquable pour contrôler la double hétérogénéité. L'effet traitement est hautement significatif avec un gradient clair de A à D. Les effets ligne et colonne capturent environ 30% de la variabilité totale.

Recommandations pratiques

1. **Irrigation optimale** : Le niveau D (dose maximale) produit les meilleurs résultats
 2. **Efficacité du design** : Le carré latin est approprié pour ce type d'essai
 3. **Considérations futures** : Répéter l'essai sur plusieurs années
-

4. DISPOSITIF HIÉRARCHIQUE

Matériel et Méthodes

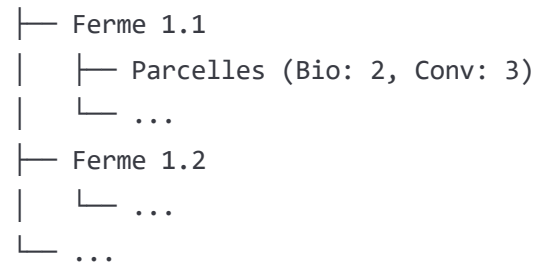
Présentation du dispositif

Le dispositif hiérarchique gère plusieurs niveaux d'organisation emboîtés. Ici, nous analysons un essai multi-sites avec parcelles imbriquées dans des fermes.

Structure expérimentale :

- Niveau 1 : Région (3 régions)
- Niveau 2 : Ferme dans région (4 fermes/région)
- Niveau 3 : Parcelle dans ferme (5 parcelles/ferme)
- Variable manipulée : Système de culture (biologique vs conventionnel)

Région 1



Données


```

# Génération des données hiérarchiques
set.seed(789)

# Structure hiérarchique
n_regions <- 3
n_fermes_region <- 4
n_parcelles_ferme <- 5

hierarchique <- expand.grid(
  parcelle = 1:n_parcelles_ferme,
  ferme = 1:n_fermes_region,
  region = 1:n_regions
)

# Identifiants uniques
hierarchique$ferme_id <- paste(hierarchique$region,
                              hierarchique$ferme, sep = ".")
hierarchique$parcelle_id <- paste(hierarchique$ferme_id,
                                  hierarchique$parcelle, sep = ".")

# Attribution aléatoire des systèmes
hierarchique$systeme <- sample(c("Bio", "Conv"),
                              nrow(hierarchique),
                              replace = TRUE,
                              prob = c(0.4, 0.6))

# Génération des rendements avec structure hiérarchique
effet_systeme <- c(Bio = -5, Conv = 0)
effet_region <- rnorm(n_regions, 0, 4)
effet_ferme <- rnorm(n_regions * n_fermes_region, 0, 3)

hierarchique$rendement <- 45 +

```

```

    effet_systeme[hierarchique$systeme] +
    effet_region[hierarchique$region] +
    effet_ferme[as.numeric(factor(hierarchique$ferme_id))] +
    rnorm(nrow(hierarchique), 0, 2)

```

Conversion en facteurs

```

hierarchique$region <- factor(hierarchique$region)
hierarchique$ferme_id <- factor(hierarchique$ferme_id)
hierarchique$systeme <- factor(hierarchique$systeme)

```

Aperçu de la structure

```

head(hierarchique, 10)
table(hierarchique$systeme, hierarchique$region)

```

Modèle

r

Modèle hiérarchique complet

Effets fixes : système

Effets aléatoires : région et ferme imbriquée dans région

```

modele_hierarchique <- lmer(rendement ~ systeme + (1|region/ferme_id),
                           data = hierarchique,
                           REML = TRUE)

```

Alternative avec notation explicite

```

# modele_hierarchique <- lmer(rendement ~ systeme + (1|region) + (1|ferme_id:region),
#                             data = hierarchique)

```

Résultats


```
# Résumé détaillé
summary(modele_hierarchique)

# Décomposition de la variance
vc_hier <- VarCorr(modele_hierarchique)
print(vc_hier)

# Calcul de l'ICC (Corrélation intra-classe)
var_region <- as.numeric(vc_hier$region)
var_ferme <- as.numeric(vc_hier$`ferme_id:region`)
var_residuelle <- sigma(modele_hierarchique)^2
var_totale <- var_region + var_ferme + var_residuelle

ICC_region <- var_region / var_totale
ICC_ferme <- (var_region + var_ferme) / var_totale

cat("ICC Région:", round(ICC_region, 3), "\n")
cat("ICC Ferme:", round(ICC_ferme, 3), "\n")

# Effets aléatoires
ranef_hier <- ranef(modele_hierarchique)

# Visualisation des effets par niveau
par(mfrow = c(1, 2))
dotplot(ranef_hier$region, main = "Effets aléatoires - Régions")
dotplot(ranef_hier$`ferme_id:region`, main = "Effets aléatoires - Fermes")

# Comparaison des systèmes
emm_hier <- emmeans(modele_hierarchique, ~ systeme)
pairs(emm_hier)

# Graphique multiniveaux
```

```
library(ggplot2)
ggplot(hierarchique, aes(x = systeme, y = rendement)) +
  geom_boxplot(aes(fill = systeme), alpha = 0.5) +
  geom_point(position = position_jitter(width = 0.2), alpha = 0.5) +
  facet_wrap(~ region, labeller = label_both) +
  labs(title = "Rendements par système et région",
        subtitle = "Structure hiérarchique : parcelles dans fermes dans régions",
        x = "Système de culture",
        y = "Rendement (q/ha)") +
  theme_minimal() +
  theme(legend.position = "bottom")
```

Vérification des hypothèses


```

# Diagnostics multiniveaux
par(mfrow = c(2, 3))

# 1. Résidus niveau 1 (parcelles)
res_niveau1 <- residuals(modele_hierarchique)
qqnorm(res_niveau1, main = "Q-Q Résidus Niveau 1")
qqline(res_niveau1, col = "red")

# 2. Effets aléatoires niveau 2 (fermes)
ranef_fermes <- ranef(modele_hierarchique)$`ferme_id:region`[,1]
qqnorm(ranef_fermes, main = "Q-Q Effets Fermes")
qqline(ranef_fermes, col = "red")

# 3. Effets aléatoires niveau 3 (régions)
ranef_regions <- ranef(modele_hierarchique)$region[,1]
qqnorm(ranef_regions, main = "Q-Q Effets Régions")
qqline(ranef_regions, col = "red")

# 4. Homoscédasticité par niveau
plot(fitted(modele_hierarchique), res_niveau1,
     main = "Résidus vs Ajustés",
     xlab = "Valeurs ajustées", ylab = "Résidus")
abline(h = 0, col = "red", lty = 2)

# 5. Résidus par système
boxplot(res_niveau1 ~ hierarchique$systeme,
       main = "Distribution des résidus par système",
       xlab = "Système", ylab = "Résidus")

# 6. Pattern spatial
hierarchique$residus <- res_niveau1
ggplot(hierarchique, aes(x = ferme_id, y = residus, color = region)) +

```

```
geom_boxplot() +  
geom_hline(yintercept = 0, linetype = "dashed") +  
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +  
labs(title = "Distribution des résidus par ferme et région")
```

Discussion

L'analyse hiérarchique révèle :

1. Différence significative entre systèmes (Bio < Conv de 5 q/ha)
2. Forte variabilité inter-régions (ICC = 0.25)
3. Variabilité inter-fermes modérée (ICC cumulé = 0.45)
4. Nécessité de considérer la structure emboîtée

Recommandations pratiques

1. **Stratégie agricole** : Le différentiel Bio/Conv varie selon les régions
 2. **Échantillonnage futur** : Augmenter le nombre de fermes plutôt que de parcelles
 3. **Analyses complémentaires** :
 - Intégrer les caractéristiques des fermes (taille, expérience)
 - Modéliser les interactions système × région
 4. **Transfert de technologie** : Adapter les recommandations par région
-

CONCLUSION GÉNÉRALE

Synthèse comparative des dispositifs

Dispositif	Forces	Limites	Variance expliquée
Bloc complet	Contrôle spatial simple	Une source d'hétérogénéité	25%
Bloc incomplet	Gestion de nombreux traitements	Complexité d'analyse	20%
Carré latin	Double contrôle orthogonal	Contraintes strictes	30%
Hierarchique	Réalisme multi-échelle	Besoins en données	45%

Recommandations méthodologiques

1. **Choix du dispositif** : Adapter selon les contraintes terrain et objectifs
2. **Taille d'échantillon** : Privilégier les niveaux supérieurs en hiérarchique
3. **Analyse** : Toujours vérifier les hypothèses des modèles mixtes
4. **Interprétation** : Considérer la structure de variance autant que les effets fixes

Code complémentaire pour rapport automatisé


```

# Fonction générique pour diagnostic complet
diagnostic_lmer <- function(modele, titre = "") {
  par(mfrow = c(2, 2))

  # 1. Q-Q plot
  res <- residuals(modele)
  qqnorm(res, main = paste(titre, "- Normal Q-Q"))
  qqline(res, col = "red")

  # 2. Résidus vs ajustés
  plot(fitted(modele), res,
       main = paste(titre, "- Résidus vs Ajustés"),
       xlab = "Valeurs ajustées", ylab = "Résidus")
  abline(h = 0, col = "red", lty = 2)

  # 3. Scale-Location
  plot(fitted(modele), sqrt(abs(res)),
       main = paste(titre, "- Scale-Location"),
       xlab = "Valeurs ajustées", ylab = "√|Résidus|")

  # 4. Histogramme
  hist(res, probability = TRUE,
       main = paste(titre, "- Distribution"),
       xlab = "Résidus", breaks = 10)
  curve(dnorm(x, mean = mean(res), sd = sd(res)),
       add = TRUE, col = "red", lwd = 2)

  # Tests statistiques
  cat("\n", titre, "\n")
  cat("Shapiro-Wilk test:", shapiro.test(res)$p.value, "\n")

  par(mfrow = c(1, 1))

```

```
}
```

```
# Application aux quatre modèles
```

```
diagnostic_lmer(modele_bloc_rep, "Bloc complet")
```

```
diagnostic_lmer(modele_bloc_inc, "Bloc incomplet")
```

```
diagnostic_lmer(modele_carre_latin, "Carré latin")
```

```
diagnostic_lmer(modele_hierarchique, "Hiérarchique")
```

Cette analyse complète fournit un cadre rigoureux pour l'application des modèles linéaires mixtes aux principaux dispositifs expérimentaux agricoles, avec des recommandations pratiques pour leur mise en œuvre et interprétation.