chapitre 10

Jean-Yves Barnagaud & Olivier Gimenez

juin 2023

# ‘A:/projets/projet\_stats/Biotope\_analyse\_donnees/C10’

ATTENTION : MODIFIER LE CHEMIN D’ACCES VERS VOTRE REPERTOIRE COURANT AVANT DE LANCER CE SCRIPT

# packages utiles et options du script

## packages

# génériques  
library(reshape2)  
library(tidyr)

>   
> Attachement du package : 'tidyr'

> L'objet suivant est masqué depuis 'package:reshape2':  
>   
> smiths

library(ggplot2)  
library(patchwork)  
library(maptools)

> Le chargement a nécessité le package : sp

> Checking rgeos availability: TRUE  
> Please note that 'maptools' will be retired during 2023,  
> plan transition at your earliest convenience;  
> some functionality will be moved to 'sp'.

library(viridis)

> Le chargement a nécessité le package : viridisLite

library(sf)

> Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf\_use\_s2() is TRUE

library(mapview)  
library(leaflet)  
library("rnaturalearth")

> Support for Spatial objects (`sp`) will be deprecated in {rnaturalearth} and will be removed in a future release of the package. Please use `sf` objects with {rnaturalearth}. For example: `ne\_download(returnclass = 'sf')`

library("rnaturalearthdata")

>   
> Attachement du package : 'rnaturalearthdata'

> L'objet suivant est masqué depuis 'package:rnaturalearth':  
>   
> countries110

library(formatR)  
mapviewOptions(fgb = FALSE)  
  
# modèles mixtes  
library(lme4)

> Le chargement a nécessité le package : Matrix

>   
> Attachement du package : 'Matrix'

> Les objets suivants sont masqués depuis 'package:tidyr':  
>   
> expand, pack, unpack

# représentations graphiques  
library(PerformanceAnalytics)

> Le chargement a nécessité le package : xts

> Le chargement a nécessité le package : zoo

>   
> Attachement du package : 'zoo'

> Les objets suivants sont masqués depuis 'package:base':  
>   
> as.Date, as.Date.numeric

>   
> ################################### WARNING ###################################  
> # We noticed you have dplyr installed. The dplyr lag() function breaks how #  
> # base R's lag() function is supposed to work, which breaks lag(my\_xts). #  
> # #  
> # If you call library(dplyr) later in this session, then calls to lag(my\_xts) #  
> # that you enter or source() into this session won't work correctly. #  
> # #  
> # All package code is unaffected because it is protected by the R namespace #  
> # mechanism. #  
> # #  
> # Set `options(xts.warn\_dplyr\_breaks\_lag = FALSE)` to suppress this warning. #  
> # #  
> # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #  
> # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #  
> # dplyr from breaking base R's lag() function. #  
> ################################### WARNING ###################################

>   
> Attachement du package : 'xts'

> L'objet suivant est masqué depuis 'package:leaflet':  
>   
> addLegend

>   
> Attachement du package : 'PerformanceAnalytics'

> L'objet suivant est masqué depuis 'package:graphics':  
>   
> legend

library(ggeffects)  
library(sjPlot)

> #refugeeswelcome

# diagnostics et post-analyses  
library(DHARMa)

> This is DHARMa 0.4.6. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa')

library(questionr)  
library(broom.mixed) # odds ratios  
library(MuMIn)  
library(performance)

## répertoire courant

Attention : le répertoire de travail est changé dans l’en-tête du script Markdown.

setwd('A:/Biotope\_analyse\_donnees/C10')

## options graphiques

Thème par défaut des graphiques

theme\_set(theme\_light(base\_size = 16))

Attention : les figures sont exportées dans des chunks de code à part qui sont invisibles dans le document généré par ce script et non exécutés. Pour exécuter ces chunks de code, aller dans le script Markdown et passer ces chunks de code en *eval = T*.

## Reproductibilité des simulations

set.seed(2020)

# 10.1 Le problème des strates

### La question et les données : tendance temporelle d’infestation de la chenille processionnaire

Les données :

aqui <-  
 read.csv2(  
 "donnees/chenille\_processionnaire.csv",  
 encoding = "UTF-8"  
 )  
  
aqui$placette <- factor(aqui$placette)  
aqui$region <- factor(aqui$region)

aqui$prop\_attaq <- aqui$nbpinsattac / aqui$nbpins

summary(aqui)

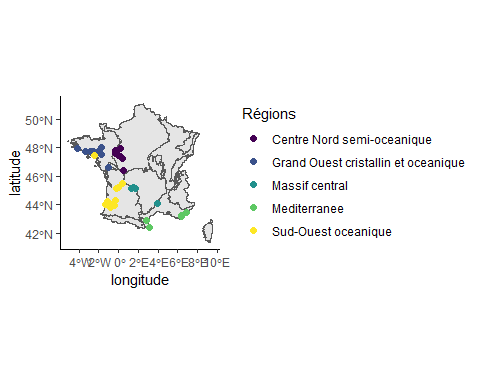
> region placette latitude   
> Centre Nord semi-oceanique :83 35051 : 22 Min. :42.44   
> Grand Ouest cristallin et oceanique:88 19011 : 19 1st Qu.:44.07   
> Massif central :43 44141 : 17 Median :46.60   
> Mediterranee :37 49211 : 17 Mean :46.02   
> Sud-Ouest oceanique :68 35131 : 16 3rd Qu.:47.72   
> 86101 : 16 Max. :48.03   
> (Other):212   
> longitude annee nbpins nbpinsattac   
> Min. :-4.11658 Min. :1986 Min. : 23.0 Min. : 0.00   
> 1st Qu.:-1.76255 1st Qu.:1991 1st Qu.: 108.0 1st Qu.: 0.00   
> Median :-0.32009 Median :1998 Median : 170.0 Median : 6.00   
> Mean :-0.03152 Mean :1998 Mean : 210.5 Mean : 18.13   
> 3rd Qu.: 0.91736 3rd Qu.:2004 3rd Qu.: 250.0 3rd Qu.: 22.00   
> Max. : 6.89048 Max. :2012 Max. :1229.0 Max. :295.00   
>   
> prop\_attaq   
> Min. :0.00000   
> 1st Qu.:0.00000   
> Median :0.04233   
> Mean :0.10491   
> 3rd Qu.:0.12899   
> Max. :0.96721   
>

Carte des placettes (données brutes)

regions <- st\_read("donnees/regions.geojson")

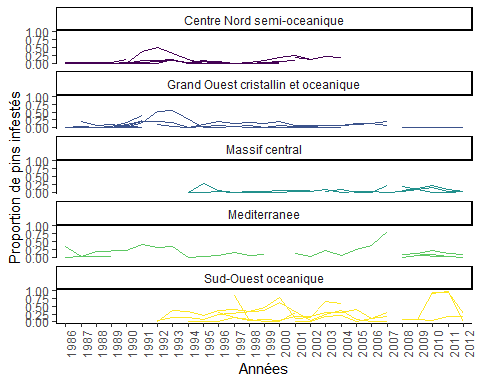
> Reading layer `regions' from data source   
> `D:\sauvegarde stats 12072023\Biotope\_analyse\_donnees\chapitres\_OK\C10\donnees\regions.geojson'   
> using driver `GeoJSON'  
> Simple feature collection with 12 features and 1 field  
> Geometry type: MULTIPOLYGON  
> Dimension: XY  
> Bounding box: xmin: -5.141328 ymin: 41.33363 xmax: 9.559955 ymax: 51.08939  
> Geodetic CRS: WGS 84

ggplot(regions) +  
 geom\_sf() +  
 geom\_point(  
 data = aqui,  
 mapping = aes(x = longitude, y = latitude, colour = region),  
 size = 2  
 ) +  
 scale\_color\_viridis(discrete = TRUE) +  
 labs(color = "Régions") +  
 theme\_classic()



Séries temporelles d’infestation par régions :

ggplot(aqui) +  
 aes(  
 x = factor(annee),  
 y = prop\_attaq,  
 group = placette,  
 color = region  
 ) +  
 geom\_line(show.legend = FALSE) +  
 labs(x = "Années", y = "Proportion de pins infestés") +  
 facet\_wrap( ~ region, ncol = 1) +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 90)) +  
 scale\_color\_viridis\_d()



On change l’échelle des années par commodité d’interprétation :

aqui$annee\_resc <- aqui$annee - min(aqui$annee) + 1

### Un premier modèle

Le GLM :

mod0 <- glm(cbind(nbpinsattac, nbpins - nbpinsattac) ~ annee\_resc, family = binomial, data = aqui)

Interprétation :

summary(mod0)

>   
> Call:  
> glm(formula = cbind(nbpinsattac, nbpins - nbpinsattac) ~ annee\_resc,   
> family = binomial, data = aqui)  
>   
> Deviance Residuals:   
> Min 1Q Median 3Q Max   
> -11.904 -4.476 -2.467 1.295 29.247   
>   
> Coefficients:  
> Estimate Std. Error z value Pr(>|z|)   
> (Intercept) -3.468608 0.030218 -114.78 <2e-16 \*\*\*  
> annee\_resc 0.080962 0.001672 48.43 <2e-16 \*\*\*  
> ---  
> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
>   
> (Dispersion parameter for binomial family taken to be 1)  
>   
> Null deviance: 15428 on 318 degrees of freedom  
> Residual deviance: 12961 on 317 degrees of freedom  
> AIC: 13922  
>   
> Number of Fisher Scoring iterations: 6

Odds ratios :

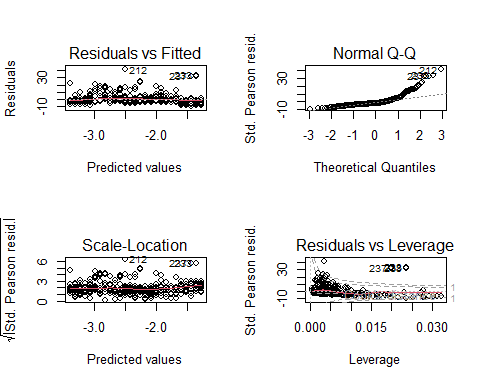
odds.ratio(mod0)

> Attente de la réalisation du profilage...

> OR 2.5 % 97.5 % p   
> (Intercept) 0.031160 0.029358 0.0331 < 2.2e-16 \*\*\*  
> annee\_resc 1.084330 1.080790 1.0879 < 2.2e-16 \*\*\*  
> ---  
> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Résidus :

par(mfrow = c(2,2))  
plot(mod0)

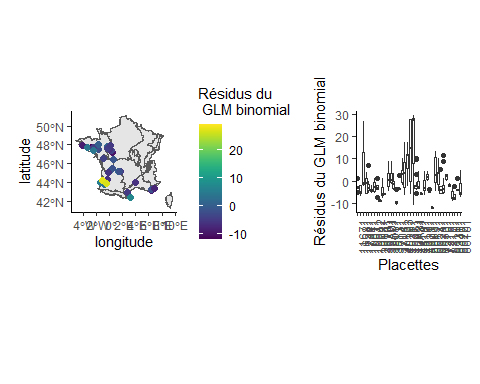


cartographie des résidus:

aqui$res.mod0 <- residuals(mod0)  
p0 <- plot(ggpredict(mod0, terms = c("annee\_resc")), residuals = T)

> Data were 'prettified'. Consider using `terms="annee\_resc [all]"` to get  
> smooth plots.

p1a <- ggplot(regions) +  
 geom\_sf() +  
 geom\_point(  
 data = aqui,  
 mapping = aes(x = longitude, y = latitude, colour = res.mod0),  
 size = 2  
 ) +  
 scale\_color\_viridis(discrete = F) +  
 labs(color = "Résidus du \n GLM binomial") +  
 theme\_classic()  
  
p1b <- ggplot(aqui) +  
 aes(x = placette, y = res.mod0) +  
 geom\_boxplot() +  
 labs(x = "Placettes", y = "Résidus du GLM binomial") +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 90))  
  
p1a + p1b

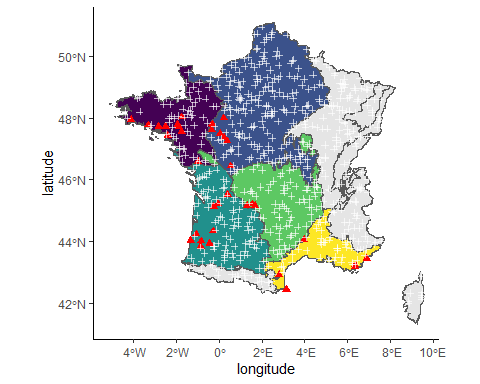


# 10.2 Représenter une stratification par un modèle mixte

### Cadre d’application et limites de l’approche

Carte : illustration de la notion d’effets aléatoires. Les placettes échantillonnées sont représentées comme des triangles rouges. Les placettes non échantillonnées sont les croix blanches.

rpoints <- st\_sample(regions, 700) %>%   
 st\_sf() %>%   
 st\_transform(4326)   
  
ggplot(regions) +  
 geom\_sf() +  
 geom\_sf(data = subset(regions, jointure\_g %in% c("A", "B", "F", "G", "J")),  
 aes(fill = jointure\_g)) +  
 geom\_point(  
 data = aqui,  
 mapping = aes(x = longitude, y = latitude),  
 colour = "red",  
 pch = 17,  
 size = 2  
 ) +  
 geom\_sf(  
 data = rpoints,  
 pch = 3,  
 col = 'white',  
 alpha = 0.67  
 ) +  
 scale\_color\_viridis(discrete = TRUE) +  
 scale\_fill\_viridis(discrete = TRUE) +  
 labs(color = "Régions") +  
 theme\_classic() +  
 theme(legend.position = "none")



### L’implémentation sous R

recodage des niveaux de facteurs par commodité d’affichage (en particulier pour la figure 17), sans impact sur le modèle.

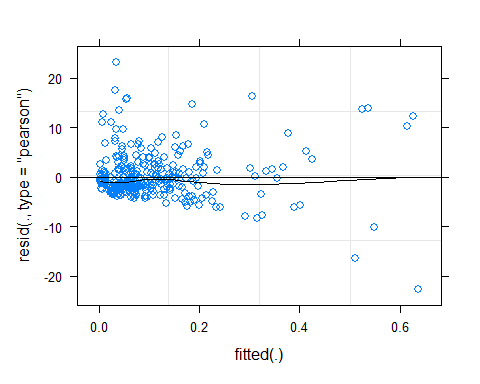
aqui$placette2 <- NA  
for (i in 1:nlevels(aqui$region)) {  
 aqui[which(aqui$region == levels(aqui$region)[i]), "placette2"] <-  
 paste(substr(levels(aqui$region)[i], 1, 1),  
 aqui[which(aqui$region == levels(aqui$region)[i]), "placette"], sep =  
 "")  
}  
  
aqui$placette2 <- factor(aqui$placette2)

Un codage possible du modèle, avec la librairie lme4. On peut aussi utiliser la librairie glmmPQL (ou la librairie mgcv, qui réexploite cette dernière). En revanche, la librairie nlme, principale alternative à lme4 pour les modèles mixtes, n’ajuste pas de GLM.

mod2 <- glmer(cbind(nbpinsattac, nbpins - nbpinsattac) ~ annee\_resc + (1|region/placette2),   
 family = binomial,   
 data = aqui)

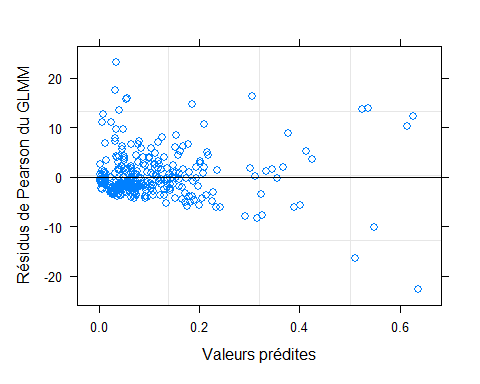
Diagnostic basique des résidus :

plot(mod2, type=c("p","smooth"), col.line=1)

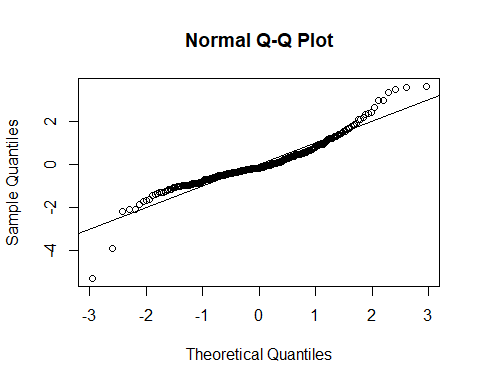


QQ plot :

plot(mod2, xlab = "Valeurs prédites", ylab = "Résidus de Pearson du GLMM")

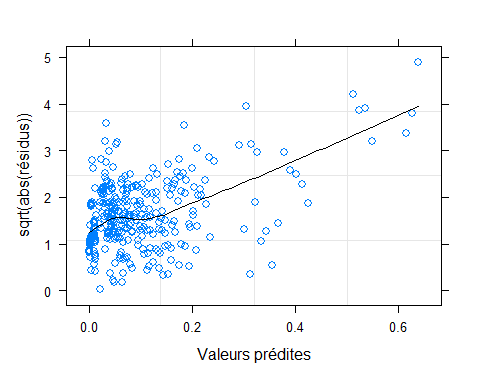


qqnorm(scale(residuals(mod2)))  
abline(0, 1)



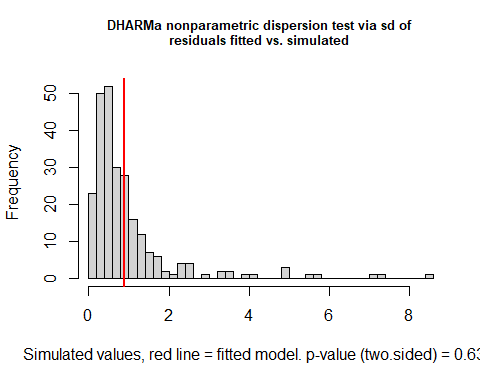
hétéroscédasticité :

plot(  
 mod2,  
 sqrt(abs(resid(.))) ~ fitted(.),  
 type = c("p", "smooth"),  
 col.line = 1,  
 xlab = "Valeurs prédites",  
 ylab = "sqrt(abs(résidus))"  
)



surdispersion :

simulationOutput <- simulateResiduals(fittedModel = mod2)  
testDispersion(simulationOutput)



>   
> DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
> simulated  
>   
> data: simulationOutput  
> dispersion = 0.92313, p-value = 0.632  
> alternative hypothesis: two.sided

### Interpréter un modèle mixte

Le résumé :

summary(mod2)

> Generalized linear mixed model fit by maximum likelihood (Laplace  
> Approximation) [glmerMod]  
> Family: binomial ( logit )  
> Formula: cbind(nbpinsattac, nbpins - nbpinsattac) ~ annee\_resc + (1 |   
> region/placette2)  
> Data: aqui  
>   
> AIC BIC logLik deviance df.resid   
> 7477.8 7492.9 -3734.9 7469.8 315   
>   
> Scaled residuals:   
> Min 1Q Median 3Q Max   
> -22.6995 -2.2075 -0.8919 1.6497 23.2264   
>   
> Random effects:  
> Groups Name Variance Std.Dev.  
> placette2:region (Intercept) 2.2736 1.5078   
> region (Intercept) 0.4732 0.6879   
> Number of obs: 319, groups: placette2:region, 38; region, 5  
>   
> Fixed effects:  
> Estimate Std. Error z value Pr(>|z|)   
> (Intercept) -3.698954 0.402895 -9.181 <2e-16 \*\*\*  
> annee\_resc 0.049029 0.004051 12.103 <2e-16 \*\*\*  
> ---  
> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
>   
> Correlation of Fixed Effects:  
> (Intr)  
> annee\_resc -0.148

Les odds-ratios, avec un intervalle de confiance de Wald :

tidy(mod2, conf.int = TRUE, exponentiate = TRUE, effects = "fixed", conf.method = "Wald")

> # A tibble: 2 × 8  
> effect term estimate std.error statistic p.value conf.low conf.high  
> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
> 1 fixed (Intercept) 0.0247 0.00997 -9.18 4.27e-20 0.0112 0.0545  
> 2 fixed annee\_resc 1.05 0.00425 12.1 1.02e-33 1.04 1.06

Les odds-ratios, avec un intervalle de confiance par profilage :

tidy(mod2, conf.int = TRUE, exponentiate = TRUE, effects = "fixed", conf.method = "profile")

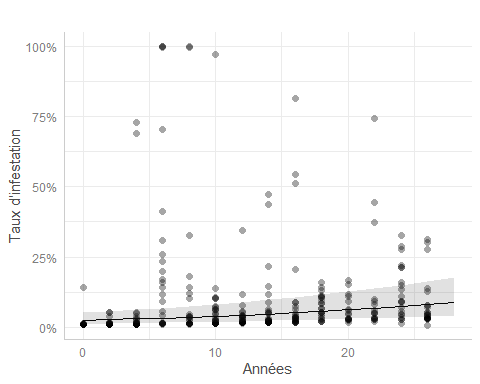
> Computing profile confidence intervals ...

> # A tibble: 2 × 8  
> effect term estimate std.error statistic p.value conf.low conf.high  
> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
> 1 fixed (Intercept) 0.0247 0.00997 -9.18 4.27e-20 0.00951 0.0629  
> 2 fixed annee\_resc 1.05 0.00425 12.1 1.02e-33 1.04 1.06

Effet prédit des années (l’argument jitter = 0 évite un sur-étalement artefactuel des points prévu par ggpredict pour éviter les empilements de données sur le graphique) :

plot(ggpredict(mod2, terms = "annee\_resc"),  
 residuals = T,  
 jitter = 0) + labs(x = "Années",  
 y = "Taux d'infestation", title = "")

> Data were 'prettified'. Consider using `terms="annee\_resc [all]"` to get  
> smooth plots.



répartition des placettes par région

res.aqui <- unique(aqui[, c("placette", "region")])  
tapply(res.aqui$placette, INDEX = res.aqui$region, FUN = "length")

> Centre Nord semi-oceanique Grand Ouest cristallin et oceanique   
> 8 9   
> Massif central Mediterranee   
> 5 6   
> Sud-Ouest oceanique   
> 10

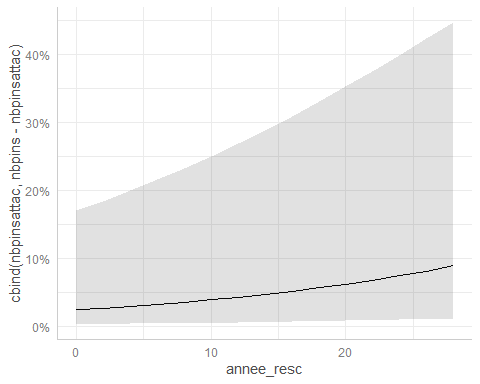
Effet de l’incertitude sur les effets aléatoires :

p1 <- ggpredict(mod2,type = "re")

> Data were 'prettified'. Consider using `terms="annee\_resc [all]"` to get  
> smooth plots.

plot(p1,  
 col = "bw",  
 show.title = F)

> $annee\_resc

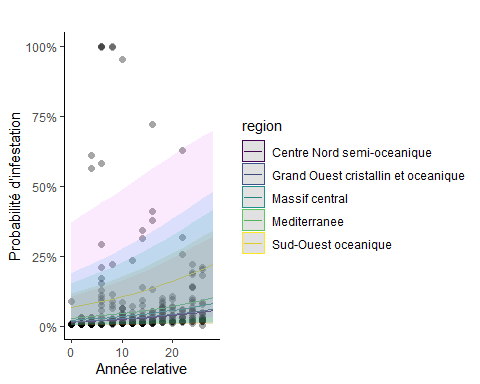


Impact des effets aléatoires sur la tendance temporelle:

plot(ggpredict(  
 mod2,  
 terms = c("annee\_resc", "region"),  
 type = "re"  
),  
residuals = T,  
jitter = F) +  
 scale\_color\_viridis\_d() +  
 theme\_classic() +  
 labs(x = "Année relative", y = "Probabilité d'infestation",title="")

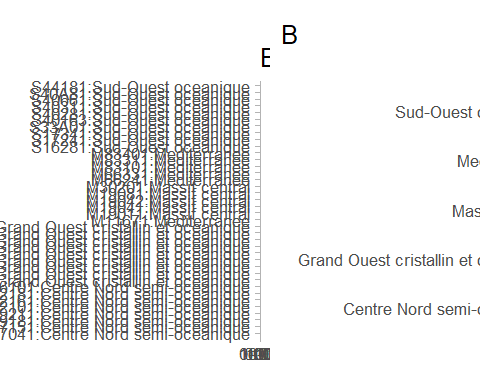
> Data were 'prettified'. Consider using `terms="annee\_resc [all]"` to get  
> smooth plots.

> Scale for colour is already present.  
> Adding another scale for colour, which will replace the existing scale.



Effets aléatoires et leur incertitude - il s’agit d’une prédiction a posteriori :

p1 <-  
 plot\_model(mod2,  
 type = "re",  
 col = "bw",  
 show.values = TRUE,  
 title = "Effet aléatoire Placettes")[[1]]  
p2 <-  
 plot\_model(mod2,  
 type = "re",  
 col = "bw",  
 show.values = TRUE,  
 title = "Effet aléatoire Régions")[[2]]  
  
p1 + p2 + plot\_annotation(tag\_levels = "A")



Estimation du R² :

r.squaredGLMM(mod2)

> Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.

> Warning: the null model is correct only if all variables used by the original  
> model remain unchanged.

> R2m R2c  
> theoretical 0.04963904 0.9946232  
> delta 0.04910875 0.9839977

# pentes aléatoires : exemple des tétras

### Variabilité d’une courbe entre strates : variations inter-massifs de la reproduction du tétras lyre dans les Alpes

tetras <- read.csv2("donnees/tetras\_lyre.csv", dec = ".")  
tetras$prop <- tetras$Nichees / tetras$Poules  
tetras$UN <- factor(tetras$UN)  
tetras$RN <- factor(tetras$RN)

Le modèle du chapitre 9

tetras.mod.quad.int <- glm(cbind(Nichees, Poules - Nichees) ~ poly(NAOdjfm, 2, raw = TRUE) \*  
 RN, family = "binomial", data = tetras)

Modèle avec effets aléatoires emboités sur l’ordonnée à l’origine : même type de modèle qu’avec les chenilles

glmm.tetras <- glmer(cbind(Nichees, Poules - Nichees) ~ poly(NAOdjfm, 2, raw = TRUE) + (1|RN/UN),   
 family = "binomial",   
 data = tetras)

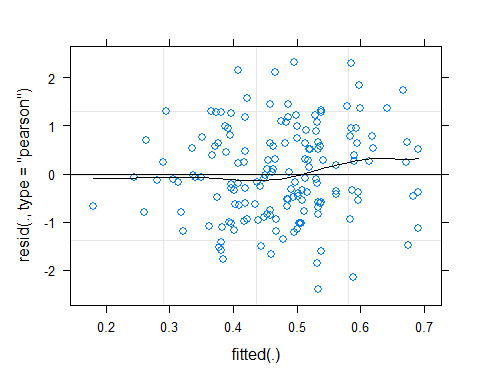
le modèle à pentes aléatoires

glmm.tetras.slope <-  
 glmer(  
 cbind(Nichees, Poules - Nichees) ~ poly(NAOdjfm, 2, raw = TRUE) + (poly(NAOdjfm, 2, raw = TRUE) |  
 RN / UN),  
 family = "binomial",  
 data = tetras  
 )

> boundary (singular) fit: see help('isSingular')

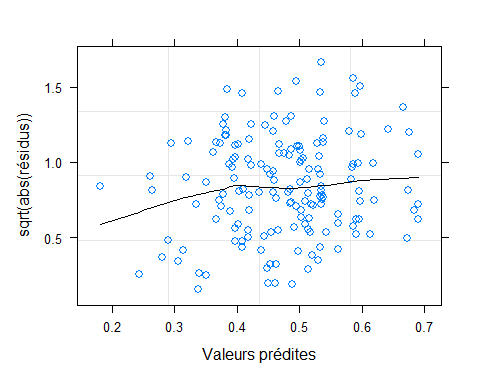
résidus

plot(glmm.tetras.slope,  
 type = c("p", "smooth"),  
 col.line = 1)



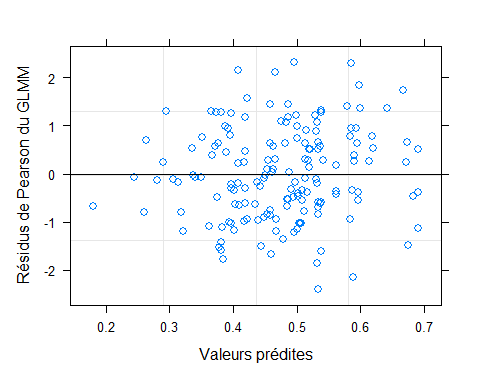
graphique suivant

plot(  
 glmm.tetras.slope,  
 sqrt(abs(resid(.))) ~ fitted(.),  
 type = c("p", "smooth"),  
 col.line = 1,  
 xlab = "Valeurs prédites",  
 ylab = "sqrt(abs(résidus))"  
)

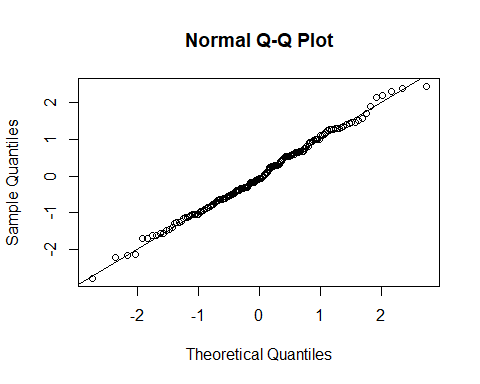


graphique suivant

plot(glmm.tetras.slope, xlab = "Valeurs prédites", ylab = "Résidus de Pearson du GLMM")

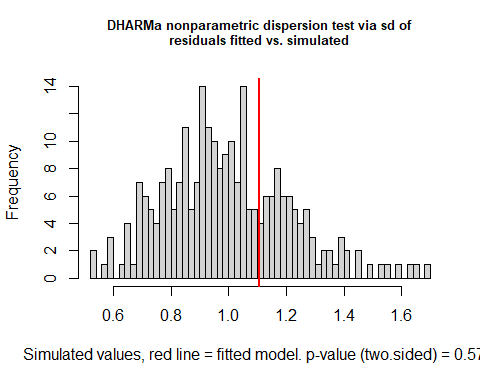


qqnorm(scale(residuals(glmm.tetras.slope)))  
abline(0, 1)



surdispersion

simulationOutput <-  
 simulateResiduals(fittedModel = glmm.tetras.slope)  
testDispersion(simulationOutput)



>   
> DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
> simulated  
>   
> data: simulationOutput  
> dispersion = 1.1122, p-value = 0.576  
> alternative hypothesis: two.sided

pour comparaison : sans interaction ni effet aléatoire

glm.tetras <-  
 glm(  
 cbind(Nichees, Poules - Nichees) ~ poly(NAOdjfm, 2, raw = TRUE) +  
 RN,  
 family = "binomial",  
 data = tetras  
 )

pour comparaison : sans aucun effet hiérarchique

glm.null <-  
 glm(  
 cbind(Nichees, Poules - Nichees) ~ poly(NAOdjfm, 2, raw = TRUE),  
 family = "binomial",  
 data = tetras  
 )

interprétation

summary(glmm.tetras)

> Generalized linear mixed model fit by maximum likelihood (Laplace  
> Approximation) [glmerMod]  
> Family: binomial ( logit )  
> Formula: cbind(Nichees, Poules - Nichees) ~ poly(NAOdjfm, 2, raw = TRUE) +   
> (1 | RN/UN)  
> Data: tetras  
>   
> AIC BIC logLik deviance df.resid   
> 627.9 643.4 -308.9 617.9 159   
>   
> Scaled residuals:   
> Min 1Q Median 3Q Max   
> -2.41235 -0.64587 -0.02743 0.68985 2.32196   
>   
> Random effects:  
> Groups Name Variance Std.Dev.  
> UN:RN (Intercept) 0.07547 0.2747   
> RN (Intercept) 0.08328 0.2886   
> Number of obs: 164, groups: UN:RN, 16; RN, 7  
>   
> Fixed effects:  
> Estimate Std. Error z value Pr(>|z|)   
> (Intercept) 0.009134 0.150245 0.061 0.95152   
> poly(NAOdjfm, 2, raw = TRUE)1 0.156433 0.055526 2.817 0.00484 \*\*   
> poly(NAOdjfm, 2, raw = TRUE)2 -0.178732 0.045458 -3.932 8.43e-05 \*\*\*  
> ---  
> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
>   
> Correlation of Fixed Effects:  
> (Intr) p(NAO,2,r=TRUE)1  
> p(NAO,2,r=TRUE)1 0.002   
> p(NAO,2,r=TRUE)2 -0.233 -0.512

avec pente aléatoire

summary(glmm.tetras.slope)

> Generalized linear mixed model fit by maximum likelihood (Laplace  
> Approximation) [glmerMod]  
> Family: binomial ( logit )  
> Formula: cbind(Nichees, Poules - Nichees) ~ poly(NAOdjfm, 2, raw = TRUE) +   
> (poly(NAOdjfm, 2, raw = TRUE) | RN/UN)  
> Data: tetras  
>   
> AIC BIC logLik deviance df.resid   
> 644.2 690.7 -307.1 614.2 149   
>   
> Scaled residuals:   
> Min 1Q Median 3Q Max   
> -2.39561 -0.65368 -0.07555 0.64231 2.31560   
>   
> Random effects:  
> Groups Name Variance Std.Dev. Corr   
> UN:RN (Intercept) 0.0894636 0.29910   
> poly(NAOdjfm, 2, raw = TRUE)1 0.0126419 0.11244 -1.00   
> poly(NAOdjfm, 2, raw = TRUE)2 0.0007614 0.02759 1.00 -1.00  
> RN (Intercept) 0.0709191 0.26631   
> poly(NAOdjfm, 2, raw = TRUE)1 0.0003500 0.01871 -1.00   
> poly(NAOdjfm, 2, raw = TRUE)2 0.0003491 0.01868 1.00 -1.00  
> Number of obs: 164, groups: UN:RN, 16; RN, 7  
>   
> Fixed effects:  
> Estimate Std. Error z value Pr(>|z|)   
> (Intercept) 0.01261 0.14763 0.085 0.93194   
> poly(NAOdjfm, 2, raw = TRUE)1 0.17859 0.06747 2.647 0.00812 \*\*   
> poly(NAOdjfm, 2, raw = TRUE)2 -0.18805 0.04767 -3.945 7.99e-05 \*\*\*  
> ---  
> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
>   
> Correlation of Fixed Effects:  
> (Intr) p(NAO,2,r=TRUE)1  
> p(NAO,2,r=TRUE)1 -0.356   
> p(NAO,2,r=TRUE)2 -0.017 -0.554   
> optimizer (Nelder\_Mead) convergence code: 0 (OK)  
> boundary (singular) fit: see help('isSingular')

représentation comparative

p7b <-  
 plot(ggpredict(glm.null, terms = "NAOdjfm [all]"), residuals = T) +  
 labs(x = "NAO décembre - mars", y = "Probabilité de reproduction", title = "sans effet hiérarchique") +  
 scale\_y\_continuous(labels = scales::percent, lim = c(0, 1)) +  
 scale\_x\_continuous(lim = c(-1.5, 2))

> Scale for y is already present.  
> Adding another scale for y, which will replace the existing scale.

p7 <-  
 plot(ggpredict(glm.tetras, terms = "NAOdjfm [all]"), residuals = T) +  
 labs(x = "NAO décembre - mars", y = "Probabilité de reproduction", title = "effet fixe RN") +  
 scale\_y\_continuous(labels = scales::percent, lim = c(0, 1)) +  
 scale\_x\_continuous(lim = c(-1.5, 2))

> Scale for y is already present.  
> Adding another scale for y, which will replace the existing scale.

p8 <-  
 plot(ggpredict(glmm.tetras, terms = "NAOdjfm [all]"), residuals = T) +  
 labs(x = "NAO décembre - mars", y = "Probabilité de reproduction", title = "effets aléatoires RN/UN \n sur l'ordonnée à l'origine") +  
 scale\_y\_continuous(labels = scales::percent, lim = c(0, 1)) +  
 scale\_x\_continuous(lim = c(-1.5, 2))

> Scale for y is already present.  
> Adding another scale for y, which will replace the existing scale.

p9 <-  
 plot(ggpredict(glmm.tetras.slope, terms = "NAOdjfm [all]"),  
 residuals = T) +  
 labs(x = "NAO décembre - mars", y = "Probabilité de reproduction", title = "effets aléatoires RN/UN \n sur l'ordonnée à l'origine et la pente") +  
 scale\_y\_continuous(labels = scales::percent, lim = c(0, 1)) +  
 scale\_x\_continuous(lim = c(-1.5, 2))

> Scale for y is already present.  
> Adding another scale for y, which will replace the existing scale.

(p7b + p7) / (p8 + p9) + plot\_annotation(tag\_levels = "A")

> Warning: Removed 7 rows containing missing values (`geom\_point()`).

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

> Warning: Removed 11 rows containing missing values (`geom\_point()`).

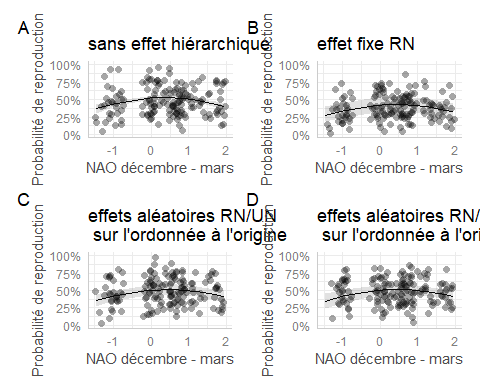
> Warning: Removed 1 row containing missing values (`geom\_line()`).

> Warning: Removed 13 rows containing missing values (`geom\_point()`).

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

> Warning: Removed 11 rows containing missing values (`geom\_point()`).

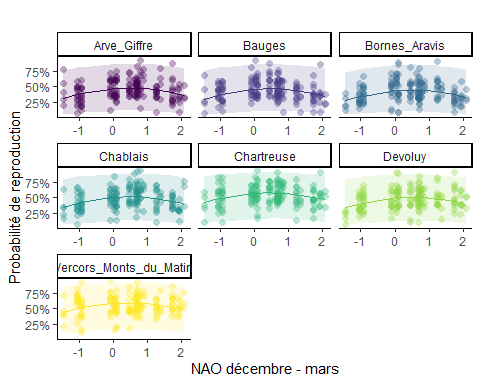
> Warning: Removed 1 row containing missing values (`geom\_line()`).



représentation des effets pentes

plot(  
 ggpredict(  
 glmm.tetras.slope,  
 terms = c("NAOdjfm[all]", "RN"),  
 type = "re"  
 ),  
 residuals = T,  
 facet = T,  
 jitter = F  
) +  
 labs(x = "NAO décembre - mars", y = "Probabilité de reproduction", title = "") +  
 scale\_color\_viridis\_d(aesthetics = c("colour", "fill"))+  
 theme\_classic()

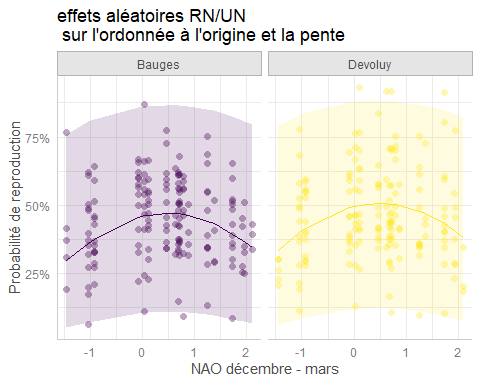
> Scale for colour is already present.  
> Adding another scale for colour, which will replace the existing scale.



pour ne représenter que quelques régions d’intérêt :

plot(  
 ggpredict(  
 glmm.tetras.slope,  
 terms = c("NAOdjfm [all]", "RN [Bauges, Devoluy]"),  
 type = "re"  
 ),  
 residuals = T,  
 facet = T,  
 jitter = F  
) +  
 labs(x = "NAO décembre - mars", y = "Probabilité de reproduction", title = "effets aléatoires RN/UN \n sur l'ordonnée à l'origine et la pente") +  
 scale\_color\_viridis\_d(aesthetics = c("colour", "fill"))

> Scale for colour is already present.  
> Adding another scale for colour, which will replace the existing scale.



### Encart 10.3. En pratique, la complexité d’un modèle mixte est-elle justifiée ?

pour répliquer la simulation à l’identique

set.seed(2023)

on crée 20 quadrats

blocs = paste("B",1:20,sep="")

l’intercept de chaque quadrat

a = abs(rnorm(20,70,10))

moyenne et erreur standard sur les 20 blocs (= ce que le modèle doit retrouver)

mean(a)

> [1] 70.15094

sd(a)/sqrt(length(a))

> [1] 1.769809

pente de chaque bloc (qu’on force à être négative)

b =rnorm(20,-5,10)

pente que le modèle doit retrouver

mean(b)

> [1] -3.610218

sd(b)/sqrt(length(b))

> [1] 2.373435

variable explicative tirée aléatoirement en chaque point (centrée réduite). On tire les mêmes valeurs dans tous les blocs

x=scale(seq(0,118,4))

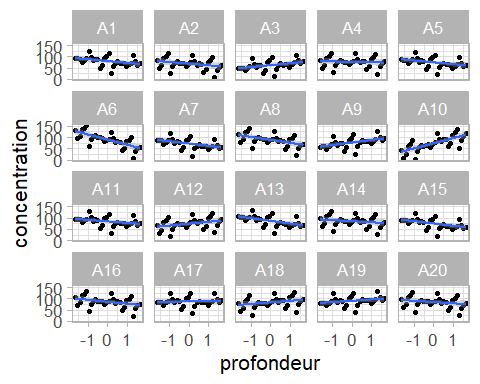
création du jeu de données

y <- matrix(NA, nrow = 30, ncol = 20)  
colnames(y) = paste("A", 1:20, sep = "")  
eps = rnorm(20, 0, 20) # erreur intra-point  
for (i in 1:20) {  
 y[, i] = a[i] + b[i] \* x  
}  
y2 = melt(y + eps)  
y2$x = rep(x, times = 20)  
colnames(y2) = c("point", "quadrat", "concentration", "profondeur")

Les données :

ggplot(y2) +  
 aes(x = profondeur, y = concentration) +  
 geom\_point() +  
 geom\_smooth(method = "lm") +  
 facet\_wrap( ~ quadrat)

> `geom\_smooth()` using formula = 'y ~ x'



Les trois modèles :

m1 = lm(concentration ~ profondeur, data = y2) # modèle linéaire sans effet bloc  
m2 = lmer(concentration ~ profondeur + (1 | quadrat), data = y2)  
m3 = lmer(concentration ~ profondeur + (profondeur |  
 quadrat), data = y2) # forte sensibilité dès que les pentes sont trop hétérogènes

paramètres :

summary(m1)

>   
> Call:  
> lm(formula = concentration ~ profondeur, data = y2)  
>   
> Residuals:  
> Min 1Q Median 3Q Max   
> -79.710 -13.995 1.557 14.204 69.411   
>   
> Coefficients:  
> Estimate Std. Error t value Pr(>|t|)   
> (Intercept) 79.4876 0.9680 82.117 < 2e-16 \*\*\*  
> profondeur -2.7451 0.9845 -2.788 0.00547 \*\*   
> ---  
> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
>   
> Residual standard error: 23.71 on 598 degrees of freedom  
> Multiple R-squared: 0.01283, Adjusted R-squared: 0.01118   
> F-statistic: 7.774 on 1 and 598 DF, p-value: 0.005469

summary(m2)

> Linear mixed model fit by REML ['lmerMod']  
> Formula: concentration ~ profondeur + (1 | quadrat)  
> Data: y2  
>   
> REML criterion at convergence: 5476.3  
>   
> Scaled residuals:   
> Min 1Q Median 3Q Max   
> -3.2968 -0.5073 0.0442 0.5500 2.8708   
>   
> Random effects:  
> Groups Name Variance Std.Dev.  
> quadrat (Intercept) 40.75 6.384   
> Residual 523.34 22.877   
> Number of obs: 600, groups: quadrat, 20  
>   
> Fixed effects:  
> Estimate Std. Error t value  
> (Intercept) 79.4876 1.7058 46.60  
> profondeur -2.7451 0.9499 -2.89  
>   
> Correlation of Fixed Effects:  
> (Intr)  
> profondeur 0.000

summary(m3)

> Linear mixed model fit by REML ['lmerMod']  
> Formula: concentration ~ profondeur + (profondeur | quadrat)  
> Data: y2  
>   
> REML criterion at convergence: 5399  
>   
> Scaled residuals:   
> Min 1Q Median 3Q Max   
> -2.71130 -0.49821 0.00143 0.43169 2.02299   
>   
> Random effects:  
> Groups Name Variance Std.Dev. Corr   
> quadrat (Intercept) 43.85 6.622   
> profondeur 97.69 9.884 -0.44  
> Residual 430.38 20.746   
> Number of obs: 600, groups: quadrat, 20  
>   
> Fixed effects:  
> Estimate Std. Error t value  
> (Intercept) 79.488 1.706 46.597  
> profondeur -2.745 2.372 -1.157  
>   
> Correlation of Fixed Effects:  
> (Intr)  
> profondeur -0.358

intervalles de confiance :

confint(m1)

> 2.5 % 97.5 %  
> (Intercept) 77.586600 81.3886894  
> profondeur -4.678599 -0.8115118

confint(m2)

> Computing profile confidence intervals ...

> 2.5 % 97.5 %  
> .sig01 3.739121 9.6409061  
> .sigma 21.602093 24.2381001  
> (Intercept) 76.065990 82.9092995  
> profondeur -4.608306 -0.8818043

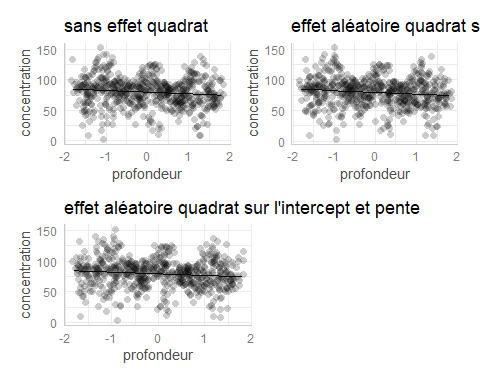
confint(m3)

> Computing profile confidence intervals ...

> 2.5 % 97.5 %  
> .sig01 4.1336605 9.79839161  
> .sig02 -0.7991843 0.08731645  
> .sig03 6.7867648 14.08945968  
> .sigma 19.5874222 22.02245188  
> (Intercept) 76.0659979 82.90929205  
> profondeur -7.5029315 2.01282132

courbes prédites :

p1 = plot(  
 ggpredict(m1, terms = "profondeur"),  
 add.data = T,  
 colors = "darkblue",  
 dot.alpha = 0.2  
) +  
 labs(title = "sans effet quadrat")  
  
p2 = plot(  
 ggpredict(m2, terms = "profondeur"),  
 add.data = T,  
 colors = "darkblue",  
 dot.alpha = 0.2  
) +  
 labs(title = "effet aléatoire quadrat sur l'intercept")  
  
p3 = plot(  
 ggpredict(m3, terms = "profondeur"),  
 add.data = T,  
 colors = "darkblue",  
 dot.alpha = 0.2  
) +  
 labs(title = "effet aléatoire quadrat sur l'intercept et pente")  
  
(p1 + p2) / (p3 + plot\_spacer())



# 10.5 Le cas des échantillonnages à stratifications croisées

### Un cas de stratification croisée : modéliser les relations abondance – paysage chez plusieurs espèces

### Les données

les données de points d’écoute :

couesnon <- read.csv2("donnees/avifaune\_couesnon.csv")  
couesnon$maille <- factor(couesnon$maille)  
couesnon$espece <- factor(couesnon$espece)  
couesnon$point <- factor(couesnon$point)  
summary(couesnon)

> maille id\_data point espece   
> A\_1 : 36 Min. : 1.0 001A\_1 : 9 accenteur\_mouchet :104   
> A\_143 : 36 1st Qu.: 454.8 001B\_1 : 9 corneille\_noire :104   
> A\_16 : 36 Median : 916.5 001C\_1 : 9 etourneau\_sansonnet :104   
> A\_166 : 36 Mean :1040.5 001D\_1 : 9 grive\_draine :104   
> A\_182 : 36 3rd Qu.:1598.2 016A\_1 : 9 mesange\_a\_longue\_queue:104   
> A\_199 : 36 Max. :2279.0 016B\_1 : 9 pic\_epeiche :104   
> (Other):720 (Other):882 (Other) :312   
> total heterogeneite configuration prairies   
> Min. : 0.000 Min. :-1.5637 Min. :-2.12397 Min. :-2.00016   
> 1st Qu.: 0.000 1st Qu.:-0.8845 1st Qu.:-0.80641 1st Qu.:-0.92869   
> Median : 0.000 Median :-0.1921 Median : 0.08543 Median :-0.08109   
> Mean : 1.473 Mean : 0.0000 Mean : 0.00000 Mean : 0.00000   
> 3rd Qu.: 2.000 3rd Qu.: 0.6963 3rd Qu.: 0.78702 3rd Qu.: 0.82344   
> Max. :28.000 Max. : 1.9088 Max. : 1.74784 Max. : 1.83367   
>   
> haies periode   
> Min. :-2.0516 Min. :1   
> 1st Qu.:-0.7048 1st Qu.:1   
> Median :-0.1723 Median :1   
> Mean : 0.0000 Mean :1   
> 3rd Qu.: 0.7846 3rd Qu.:1   
> Max. : 2.1897 Max. :1   
>

la liste des espèces :

levels(couesnon$espece)

> [1] "accenteur\_mouchet" "corneille\_noire" "etourneau\_sansonnet"   
> [4] "grive\_draine" "mesange\_a\_longue\_queue" "pic\_epeiche"   
> [7] "pinson\_des\_arbres" "pouillot\_veloce" "tourterelle\_des\_bois"

carte des points :

geojson <- st\_read("donnees/mailles\_landbio.geojson")

> Reading layer `mailles\_landbio' from data source   
> `D:\sauvegarde stats 12072023\Biotope\_analyse\_donnees\chapitres\_OK\C10\donnees\mailles\_landbio.geojson'   
> using driver `GeoJSON'  
> Simple feature collection with 31 features and 2 fields  
> Geometry type: MULTIPOLYGON  
> Dimension: XY  
> Bounding box: xmin: -1.648326 ymin: 48.25573 xmax: -1.053819 ymax: 48.60445  
> Geodetic CRS: WGS 84

p1 <-  
 mapview(  
 geojson,  
 color = "black",  
 col.regions = "#fde725",legend = F, map.types="OpenStreetMap"  
 )

mapshot(p1,file="outputs/C10F22.png")

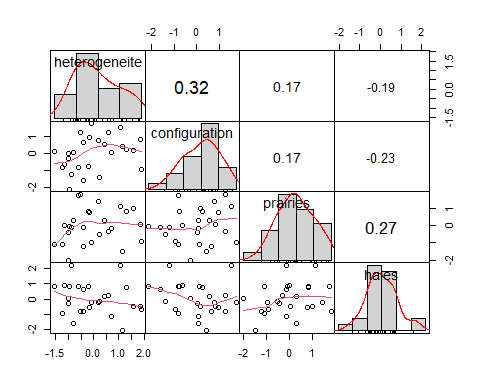
pas dans l’ouvrage car fond de carte non libre de droits : pour afficher un fond satellite :

villes <- data.frame(  
 ville = c("Rennes", "Fougères", "Combourg", "Dol de Bretagne"),  
 latitude = c(48.1173, 48.3538, 48.4110, 48.5486),  
 longitude = c(-1.6778, -1.2045, -1.7464, -1.7499)  
)  
villes\_sp <- sf::st\_as\_sf(villes, coords = c("longitude", "latitude"))  
  
p1 <-  
 mapview(  
 geojson,  
 color = "black",  
 col.regions = "#fde725",legend = F, map.types="OpenStreetMap"  
 )+ mapview(villes\_sp,color = "white", col.regions = "black",legend = F,alpha = 1,label = villes\_sp$ville,labelOptions = labelOptions(noHide = TRUE, textOnly = TRUE))

corrélation des variables explicatives (échelle mailles)

couesnon.maille <- unique(couesnon[,c("maille","heterogeneite","configuration","prairies","haies")])  
couesnon.cor <- couesnon.maille[,-1]  
chart.Correlation(couesnon.cor, histogram = T, pch= 19)

> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter



png(filename = "outputs/C10F24.png",width = 20,height = 20,res=300,units = "cm",bg="white")  
chart.Correlation(couesnon.cor, histogram = T, pch= 19)

> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter  
  
> Warning in par(usr): argument 1 does not name a graphical parameter

dev.off()

> png   
> 2

moyenne et variance des nombres d’oiseaux par point :

mean(couesnon$total)

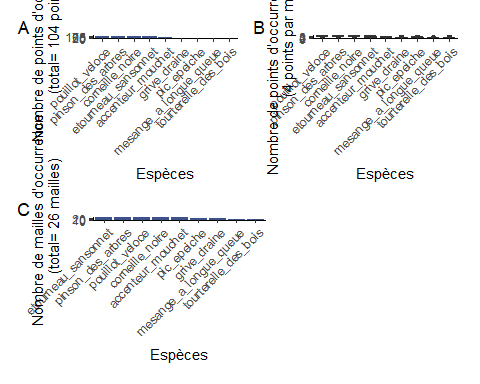
> [1] 1.473291

var(couesnon$total)

> [1] 6.373617

nombre de points et de mailles d’occurrence de chaque espèce :

# nombre de points d'occurrence  
  
df.sp <- data.frame(occ = tapply(  
 couesnon$total,  
 INDEX = couesnon$espece,  
 FUN = function(x) {  
 sum(x > 0)  
 }  
))  
df.sp$espece <- rownames(df.sp)  
  
p1 <- ggplot(df.sp) +  
 aes(x = reorder(espece, -occ), y = occ) +  
 geom\_bar(stat = "identity", fill = "#3b528b") +  
 theme\_classic() +  
 labs(x = "Espèces", y = "Nombre de points d'occurrence \n (total= 104 points)") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 theme(text = element\_text(size = 11))  
  
# nombre de points d'occurrence par maille  
  
df.sp2 <- aggregate(  
 couesnon$total,  
 by = list(couesnon$maille, couesnon$espece),  
 FUN = function(x) {  
 sum(x > 0)  
 }  
)  
colnames(df.sp2) <- c("maille","espece","n\_points\_mailles")  
  
  
p2 <- ggplot(df.sp2) +  
 aes(x = reorder(espece, -n\_points\_mailles), y = n\_points\_mailles) +  
 geom\_boxplot() +  
 theme\_classic() +  
 labs(x = "Espèces", y = "Nombre de points d'occurrence par maille \n (4 points par maille") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 theme(text = element\_text(size = 11)) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 theme(text = element\_text(size = 11))  
  
# nombre de mailles d'occurrence  
  
df.sp3 <-  
 aggregate(couesnon$total,  
 by = list(couesnon$espece, couesnon$maille),  
 FUN = sum)  
colnames(df.sp3) <- c("espece","maille","total")  
  
df.sp4 <- data.frame(occ = tapply(  
 df.sp3$total,  
 INDEX = df.sp3$espece,  
 FUN = function(x) {  
 sum(x > 0)  
 }  
))  
df.sp4$espece <- rownames(df.sp4)  
  
p3 <- ggplot(df.sp4) +  
 aes(x = reorder(espece, -occ), y = occ) +  
 geom\_bar(stat = "identity", fill = "#3b528b") +  
 theme\_classic() +  
 labs(x = "Espèces", y = "Nombre de mailles d'occurrence \n (total= 26 mailles)") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 theme(text = element\_text(size = 11))  
  
(p1 + p2) / (p3 + plot\_spacer()) + plot\_annotation(tag\_levels = "A")



### Un modèle sans stratification

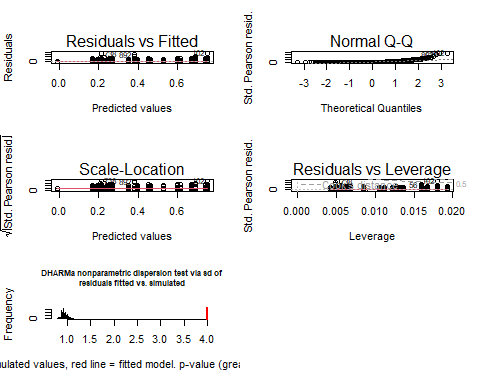
Le GLM sans aucune stratification :

glm.couesnon <-  
 glm(  
 total ~ poly(heterogeneite, 2) + poly(configuration, 2) + poly(prairies, 2) + poly(haies, 2),  
 family = "poisson",  
 data = couesnon  
 )

Les résidus:

par(mfrow = c(3, 2))  
plot(glm.couesnon)  
  
sim.glm.couesnon2 <- simulateResiduals(fittedModel = glm.couesnon)  
testDispersion(sim.glm.couesnon2, alternative = "greater")

>   
> DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
> simulated  
>   
> data: simulationOutput  
> dispersion = 4.2608, p-value < 2.2e-16  
> alternative hypothesis: greater



quelques points de résidus extrêmes : ils correspondent à des groupes d’étourneaux et aux trois seules données > 16 oiseaux

couesnon[c(102,738,892),]

> maille id\_data point espece total heterogeneite configuration  
> 102 A\_16 1055 016D\_1 etourneau\_sansonnet 28 -0.9486535 -2.12397397  
> 738 A\_682 1193 682A\_1 etourneau\_sansonnet 20 -1.0120927 0.01170838  
> 892 A\_708 1231 708D\_1 etourneau\_sansonnet 19 -0.4177988 0.63957046  
> prairies haies periode  
> 102 -0.06084474 0.8170573 1  
> 738 -0.40498546 -0.3018716 1  
> 892 -1.20718993 -0.8225650 1

subset(couesnon,total>16)

> maille id\_data point espece total heterogeneite configuration  
> 102 A\_16 1055 016D\_1 etourneau\_sansonnet 28 -0.9486535 -2.12397397  
> 738 A\_682 1193 682A\_1 etourneau\_sansonnet 20 -1.0120927 0.01170838  
> 892 A\_708 1231 708D\_1 etourneau\_sansonnet 19 -0.4177988 0.63957046  
> prairies haies periode  
> 102 -0.06084474 0.8170573 1  
> 738 -0.40498546 -0.3018716 1  
> 892 -1.20718993 -0.8225650 1

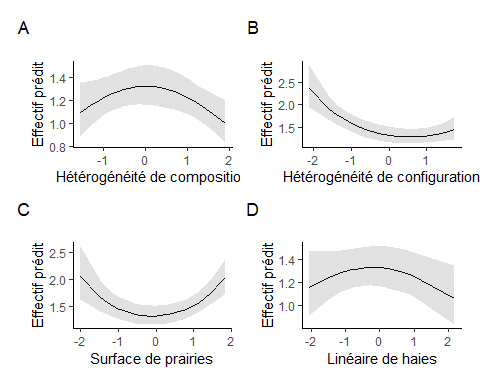
les paramètres :

summary(glm.couesnon)

>   
> Call:  
> glm(formula = total ~ poly(heterogeneite, 2) + poly(configuration,   
> 2) + poly(prairies, 2) + poly(haies, 2), family = "poisson",   
> data = couesnon)  
>   
> Deviance Residuals:   
> Min 1Q Median 3Q Max   
> -2.0612 -1.6082 -1.4082 0.5772 9.6261   
>   
> Coefficients:  
> Estimate Std. Error z value Pr(>|z|)   
> (Intercept) 0.36599 0.02749 13.313 < 2e-16 \*\*\*  
> poly(heterogeneite, 2)1 -1.00977 0.99437 -1.015 0.309875   
> poly(heterogeneite, 2)2 -2.19378 0.98522 -2.227 0.025968 \*   
> poly(configuration, 2)1 -3.45414 0.92110 -3.750 0.000177 \*\*\*  
> poly(configuration, 2)2 2.78171 0.86941 3.200 0.001376 \*\*   
> poly(prairies, 2)1 0.73295 0.88960 0.824 0.409988   
> poly(prairies, 2)2 3.99015 0.83447 4.782 1.74e-06 \*\*\*  
> poly(haies, 2)1 -0.84047 1.01822 -0.825 0.409126   
> poly(haies, 2)2 -1.64025 0.88871 -1.846 0.064944 .   
> ---  
> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
>   
> (Dispersion parameter for poisson family taken to be 1)  
>   
> Null deviance: 2835.3 on 935 degrees of freedom  
> Residual deviance: 2774.4 on 927 degrees of freedom  
> AIC: 4054.8  
>   
> Number of Fisher Scoring iterations: 6

on peut regarder les courbes :

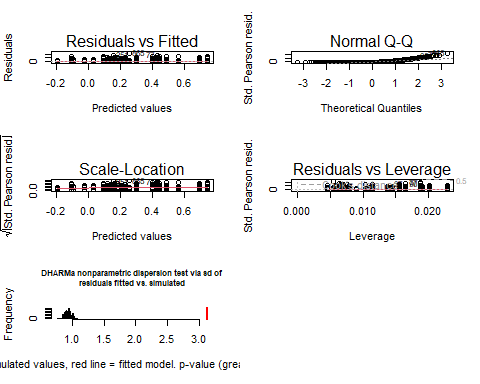
p1 <- plot(ggpredict(glm.couesnon, terms = "heterogeneite[all]"))+  
 labs(x = "Hétérogénéité de composition", y = "Effectif prédit", title ="")+  
 theme\_classic()  
p2 <- plot(ggpredict(glm.couesnon, terms = "configuration[all]"))+  
 labs(x = "Hétérogénéité de configuration", y = "Effectif prédit", title ="")+  
 theme\_classic()  
p3 <- plot(ggpredict(glm.couesnon, terms = "prairies[all]"))+  
 labs(x = "Surface de prairies", y = "Effectif prédit", title ="")+  
 theme\_classic()  
p4 <- plot(ggpredict(glm.couesnon, terms = "haies[all]"))+  
 labs(x = "Linéaire de haies", y = "Effectif prédit", title ="")+  
 theme\_classic()  
  
(p1 + p2) / (p3 + p4) + plot\_annotation(tag\_levels = "A")



analyse de sensibilité : le même modèle sans l’étourneau. C’est mieux, mais pas parfait : il doit y avoir d’autres espèces surabondantes ou des points ou mailles à fortes abondances. Nous pourrions poursuivre l’investigation (il le faudrait en cas réel), mais pour cet exemple nous avons surtout besoin de réaliser que l’identité de l’espèce est probablement en partie responsable de la surdispersion.

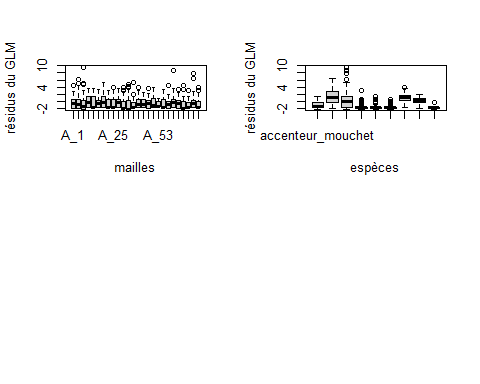
couesnon2 <- subset(couesnon, espece!= "etourneau\_sansonnet")  
  
glm.couesnon2 <-  
 glm(  
 total ~ poly(heterogeneite, 2) + poly(configuration, 2) + poly(prairies, 2) + poly(haies, 2),  
 family = poisson,  
 data = couesnon2  
 )  
  
sim.glm.couesnon2 <- simulateResiduals(fittedModel = glm.couesnon2)  
  
par(mfrow = c(3, 2))  
plot(glm.couesnon2)  
testDispersion(sim.glm.couesnon2, alternative = "greater")

>   
> DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
> simulated  
>   
> data: simulationOutput  
> dispersion = 3.3736, p-value < 2.2e-16  
> alternative hypothesis: greater



Variation résiduelle par mailles et par espèces

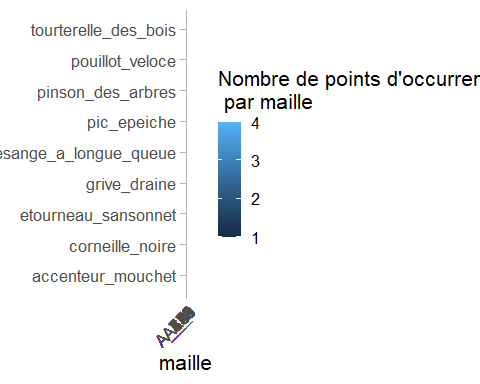
par(mfrow = c(2, 2))  
  
boxplot(residuals(glm.couesnon) ~ couesnon$maille,  
 xlab = "mailles",  
 ylab = "résidus du GLM")  
  
boxplot(residuals(glm.couesnon) ~ couesnon$espece,  
 xlab = "espèces",  
 ylab = "résidus du GLM")



### Un modèle mixte à effets aléatoires croisés : stratifications par mailles et par espèces

visualiser la stratification croisée

ggplot(df.sp2) +  
 aes(  
 x = maille,  
 y = espece,  
 fill = ifelse(n\_points\_mailles > 0, n\_points\_mailles, NA)  
 ) +  
 scale\_fill\_continuous(name = "Nombre de points d'occurrence \n par maille", na.value = 'white') +  
 geom\_tile() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



le modèle :

glmm.couesnon <-  
 glmer(  
 total ~ poly(heterogeneite, 2) + poly(configuration, 2) + poly(prairies, 2) + poly(haies, 2) +  
 (1 | maille) + (1 | espece),  
 family = "poisson",  
 data = couesnon  
 )

résidus avec la librairie performance :

check\_model(glmm.couesnon)

on voit nettement l’impact de la stratification sur l’incertitude des paramètres

summary(glmm.couesnon)

> Generalized linear mixed model fit by maximum likelihood (Laplace  
> Approximation) [glmerMod]  
> Family: poisson ( log )  
> Formula:   
> total ~ poly(heterogeneite, 2) + poly(configuration, 2) + poly(prairies,   
> 2) + poly(haies, 2) + (1 | maille) + (1 | espece)  
> Data: couesnon  
>   
> AIC BIC logLik deviance df.resid   
> 2601.8 2655.0 -1289.9 2579.8 925   
>   
> Scaled residuals:   
> Min 1Q Median 3Q Max   
> -2.5074 -0.5611 -0.3335 0.2798 11.2355   
>   
> Random effects:  
> Groups Name Variance Std.Dev.  
> maille (Intercept) 0.03824 0.1955   
> espece (Intercept) 2.36879 1.5391   
> Number of obs: 936, groups: maille, 26; espece, 9  
>   
> Fixed effects:  
> Estimate Std. Error z value Pr(>|z|)   
> (Intercept) -0.52771 0.51897 -1.017 0.30923   
> poly(heterogeneite, 2)1 -0.74518 1.66987 -0.446 0.65542   
> poly(heterogeneite, 2)2 -2.94816 1.69439 -1.740 0.08187 .   
> poly(configuration, 2)1 -3.62232 1.64679 -2.200 0.02783 \*   
> poly(configuration, 2)2 2.25212 1.58592 1.420 0.15559   
> poly(prairies, 2)1 0.08328 1.64632 0.051 0.95966   
> poly(prairies, 2)2 4.31872 1.52420 2.833 0.00461 \*\*  
> poly(haies, 2)1 -0.14469 1.76296 -0.082 0.93459   
> poly(haies, 2)2 -1.86039 1.49784 -1.242 0.21422   
> ---  
> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
>   
> Correlation of Fixed Effects:  
> (Intr) ply(ht,2)1 ply(ht,2)2 ply(c,2)1 ply(c,2)2 ply(p,2)1  
> ply(htr,2)1 0.000   
> ply(htr,2)2 0.004 -0.123   
> ply(cnf,2)1 0.003 -0.288 0.207   
> ply(cnf,2)2 -0.001 -0.013 0.313 -0.005   
> ply(prr,2)1 0.001 -0.217 0.328 -0.102 0.106   
> ply(prr,2)2 -0.005 0.260 -0.060 -0.252 0.107 -0.090   
> poly(hs,2)1 -0.001 0.225 -0.189 0.166 -0.322 -0.385   
> poly(hs,2)2 0.002 -0.126 0.014 -0.057 0.029 0.106   
> ply(p,2)2 ply(hs,2)1  
> ply(htr,2)1   
> ply(htr,2)2   
> ply(cnf,2)1   
> ply(cnf,2)2   
> ply(prr,2)1   
> ply(prr,2)2   
> poly(hs,2)1 0.138   
> poly(hs,2)2 -0.031 -0.080

R² : il est très faible, suggérant que l’effet des variables paysagères est très bruité. Ce n’est pas forcément étonnant, car les variables explicatives ne sont pas mesurées à la même échelle que les abondances d’oiseaux. On pourrait réfléchir à un moyen de modéliser l’abondance des oiseaux à l’échelle de la maille, et estimer l’effet des variables paysagères sur ces abondances à la maille. C’est faisable à l’aide de modèles hiérarchiques mais nous n’irons pas jusque là.

r.squaredGLMM(glmm.couesnon)

> Warning: the null model is correct only if all variables used by the original  
> model remain unchanged.

> R2m R2c  
> delta 0.01557404 0.8317557  
> lognormal 0.01608263 0.8589180  
> trigamma 0.01485578 0.7933960

(Pas dans l’ouvrage) On peut néanmoins tenter une amélioration en structurant les effets des variables par espèces (pentes aléatoires). Attention, pas d’effet aléatoire maille sur les pentes : ça n’aurait pas de sens puisqu’il n’y a pas de variabilité des variables explicatives à l’intérieur des mailles. Ce modèle commence à être un peu long à tourner et va poser des problèmes de convergence à cause d’espèces insuffisamment échantillonnées le long des gradients d’intérêt : nous en resterons donc là.

p1 <- plot(ggpredict(glmm.couesnon, terms = "heterogeneite[all]"),residuals=T)+  
 labs(x = "Hétérogénéité de composition", y = "Effectif prédit", title ="")+  
 theme\_classic()+  
 ylim(0,10)

> Scale for y is already present.  
> Adding another scale for y, which will replace the existing scale.

p2 <- plot(ggpredict(glmm.couesnon, terms = "configuration[all]"),residuals=T)+  
 labs(x = "Hétérogénéité de configuration", y = "Effectif prédit", title ="")+  
 theme\_classic()+  
 ylim(0,10)

> Scale for y is already present.  
> Adding another scale for y, which will replace the existing scale.

p3 <- plot(ggpredict(glmm.couesnon, terms = "prairies[all]"),residuals=T)+  
 labs(x = "Surface de prairies", y = "Effectif prédit", title ="")+  
 theme\_classic()+  
 ylim(0,10)

> Scale for y is already present.  
> Adding another scale for y, which will replace the existing scale.

p4 <- plot(ggpredict(glmm.couesnon, terms = "haies[all]"),residuals=T)+  
 labs(x = "Linéaire de haies", y = "Effectif prédit", title ="")+  
 theme\_classic()+  
 ylim(0,10)

> Scale for y is already present.  
> Adding another scale for y, which will replace the existing scale.

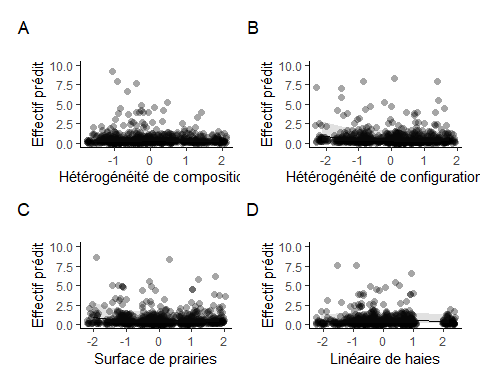
(p1 + p2) / (p3 + p4) + plot\_annotation(tag\_levels = "A")

> Warning: Removed 60 rows containing missing values (`geom\_point()`).

> Warning: Removed 46 rows containing missing values (`geom\_point()`).

> Warning: Removed 48 rows containing missing values (`geom\_point()`).

> Warning: Removed 52 rows containing missing values (`geom\_point()`).



Sans les résidus partiels : c’est plus lisible, mais ça ne doit pas faire oublier le bruit résiduel autour de ces courbes lié aux espèces mal prédites, principalement par excès de 0 ou de très grandes valeurs.

p1 <- plot(ggpredict(glmm.couesnon, terms = "heterogeneite[all]"))+  
 labs(x = "Hétérogénéité de composition", y = "Effectif prédit", title ="")+  
 theme\_classic()  
p2 <- plot(ggpredict(glmm.couesnon, terms = "configuration[all]"))+  
 labs(x = "Hétérogénéité de configuration", y = "Effectif prédit", title ="")+  
 theme\_classic()  
p3 <- plot(ggpredict(glmm.couesnon, terms = "prairies[all]"))+  
 labs(x = "Surface de prairies", y = "Effectif prédit", title ="")+  
 theme\_classic()  
p4 <- plot(ggpredict(glmm.couesnon, terms = "haies[all]"))+  
 labs(x = "Linéaire de haies", y = "Effectif prédit", title ="")+  
 theme\_classic()  
  
(p1 + p2) / (p3 + p4) + plot\_annotation(tag\_levels = "A")

