

# Rapport Modèle Bayesian Hiérarchique avec WSPROP

Antoine Habib (Service Civil)

03.06-06.12.2024

## Introduction:

Ce fichier Rmarkdown a pour but d'être un guide accompagnant la méthode développée durant le cours pour l'élaboration du modèle bayésien hiérarchique. Certaines sections du code sont incluses dans ce Rmarkdown. Les codes complets se trouvent dans trois scripts séparés.

```
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE) #wrap code text

library(pacman)
source("C:/Users/IF_240606/Desktop/Projet/functions.R")
p_load(httr, httpuv, StanHeaders, eSDM, spOccupancy, jsonlite, dplyr, data.table, readr, sf, ggplot2, )

if(file.exists("C:/Users/IF_240606/Desktop/Projet/.Renviron")){
  readRenviron("C:/Users/IF_240606/Desktop/Projet/.Renviron")
} else {
  print("No .Renviron existing!")
}

### set-up the authentication rights
client_key = Sys.getenv("client_key")
client_secret = Sys.getenv("client_secret")
file.remove(".httr-oauth")
auth.code <- httr::oauth2.0_token(endpoint = httr::oauth_endpoint(authorize="https://auth.infoflora.ch/oauth2/authorize",
                                                                    acces="https://auth.infoflora.ch/oauth2/token"),
                                app = httr::oauth_app(
                                  appname = "infoflora",
                                  key = client_key,
                                  secret = client_secret
                                ),
                                scope = "observation:read-all",
                                use_oob = FALSE, cache = TRUE, client_credentials = TRUE)
```

## Motivation du modèle

Le modèle a été développé pour que nous puissions comparer l'approche suivante avec d'autres méthodes, en particulier le "Swiss Plant Index".

La particularité de notre modèle est la prise en compte des données de l'Atlas Welten Sutter (atlas\_id : 27 pour la requête API). Je discuterai le développement de cette méthode en détail ci-dessous. De plus, nous voulions modéliser cela en intégrant deux facteurs plus généraux : l'effort d'échantillonnage et l'adéquation de l'habitat (suitability). Notre objectif était de créer un modèle spatial et dynamique (temporel) pour évaluer les populations sur une période de trente ans. Cela nous a donc poussés à utiliser le package SpOccupancy.

Cette analyse a été réalisée sur trois espèces : *Medicago lupulina* L., *Festuca valesiaca* Gaudin et *Saxifraga hirculus* L.. Chaque espèce présente des tendances de population différentes (répandue, localement rare, rare).

### 1. Monte-Carlo par chaînes de Markov:

Une façon de formuler mathématiquement le problème que nous essayons de résoudre est la suivante. Supposons que  $p(\theta)$  soit la distribution de probabilité à priori de  $\theta$  qui dans notre cas prend la forme d'un vecteur car nous avons plusieurs paramètres. De plus, si  $p(y)$  est notre probabilité marginale des données, nous pouvons exprimer le problème dans la manière suivante:  $P(y | \theta)$  ou la probabilité des observations  $y$  conditionnelle à une valeur de  $\theta$ . Dans la littérature, cela est souvent nommé "fonction de vraisemblance". Grâce au théorème Bayes,

$$P(\theta | y) = \frac{P(y | \theta)P(\theta)}{P(y)}$$

Cette formulation est puissante car  $p(y)$  peut être vue comme une constante de normalisation nécessaire pour que:

$$\int_{\theta} P(\theta | y) d\theta = 1$$

### 2. Algorithme de Metropolis-Hastings:

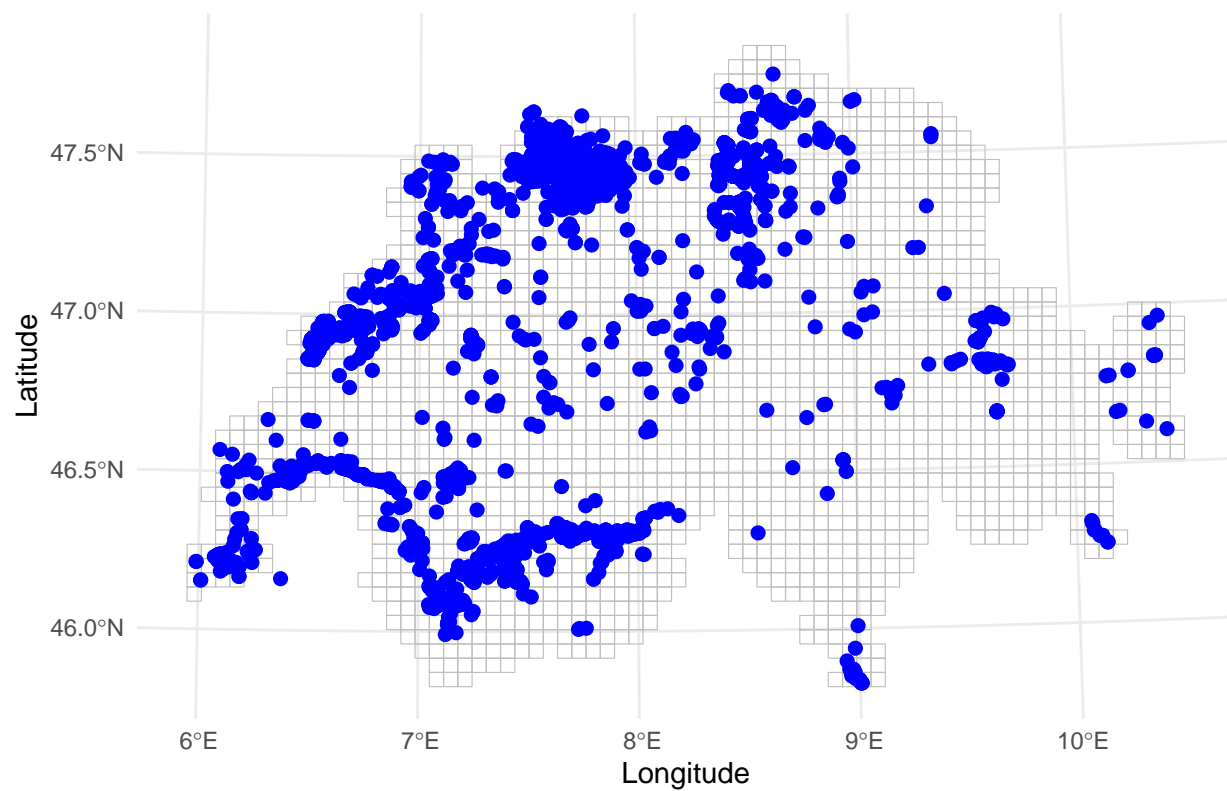
L'algorithme permet de générer un échantillon de la distribution  $p(\theta | y)$  à partir des rapports de probabilité.

L'objectif de l'algorithme de Metropolis-Hastings est de générer une chaîne de Markov dont la distribution stationnaire est la distribution cible. Une chaîne Markov aura une distribution stationnaire unique si 3 conditions sont remplies:

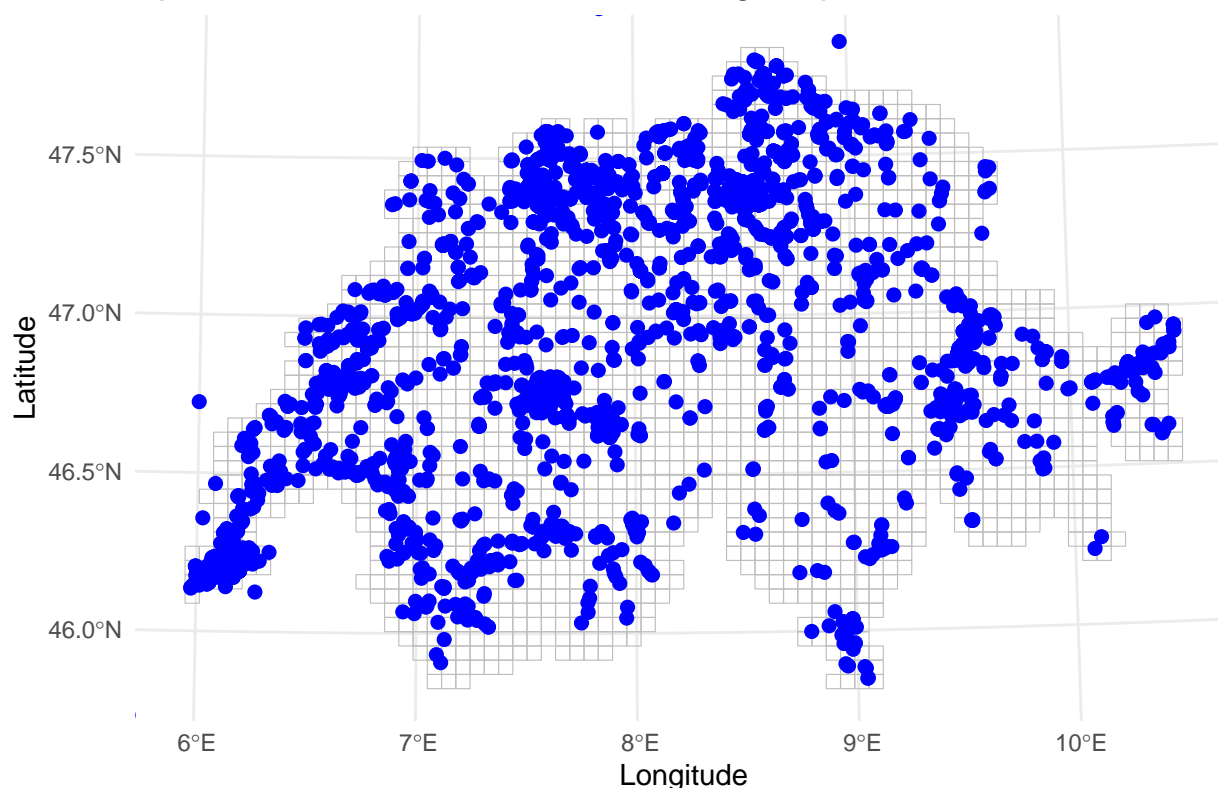
1. Irréductibilité: de pouvoir accéder n'importe quel état dans la chaîne
2. Apériodicité: La chaîne n'a pas de cycle fixe
3. Récurrence positive: Si le temps attendu de retour à chaque position dans la chaîne est fini

**3. Développement d'un modèle hiérarchique:** Nous graphons les observations de *Medicago Lupulina* L. sur la grille des présences calculer à base de ces observations.

Spatial Plot of Observations for *Medicago Lupulina* L. 1994



## Spatial Plot of Observations for Medicago Lupulina L. 2023



### L'Approche pour convertir l'information de l'atlas Welten Sutter en carrés 5x5:

L'information Welten Sutter est unique pour chaque taxon. Voici un exemple de requête pour *Medicago Lupulina*:

```
taxa = 1028530 #Medicago Lupulina L.

ws_atlas <- httr::GET(paste0("https://obs.infoflora.ch/rest/v4/communities/0/public/atlasses/27/names/",
  taxa, "/units?srid=4326&with-legend=true")) %>%
  httr::content(as = "text", encoding = "UTF-8") %>%
  jsonlite::fromJSON() %>%
  .$data #Table of all available taxa in CH (and their IDs)

head(ws_atlas)
```

```
##           id year_min year_max presence doubt intro obs_nb gis_id legend
## 1 14095246057    1982    1982         2     0     0     0    166      3
## 2 14095246292    1982    1982         2     0     0     0    135      3
## 3 14095247520    1982    1982         2     0     0     0    147      3
## 4 14095247592    1982    1982         2     0     0     0    148      3
## 5 14095249473    1982    1982         2     0     0     0    178      3
## 6 14095252994    1982    1982         2     0     0     0    176      3
##           x           y
## 1 7.340450 47.39997
## 2 6.835506 47.03722
## 3 7.011218 47.43758
```

```
## 4 7.113645 47.45385
## 5 7.674401 47.47684
## 6 7.601446 47.40729
```

Chaque secteur Welten Sutter, identifié par son identifiant (`gis_id`), est associé à une légende. Celle-ci se réfère à la fréquence qui lui a été attribuée. Dans notre analyse, nous avons regroupé toutes les fréquences qui n'étaient pas classifiées comme "fréquent" (légende 3).

À partir de cette distinction, nous avons intersecté les observations avec une grille de 5x5 km, pour traduire ces observations en présences spatiales. Ensuite, nous avons calculé la proportion de chaque secteur WS dans l'atlas qui est couvert par nos carrés de 5x5 km dans lesquels nous avons des observations.

Le code ci-dessous illustre l'approche discutée.

```
ws <- st_read("C:/Users/IF_240606/Desktop/Projet/welten_sutter_with_id.gpkg",
  layer = "welten_sutter")

## Reading layer `welten_sutter' from data source
##   `C:\Users\IF_240606\Desktop\Projet\welten_sutter_with_id.gpkg'
##   using driver `GPKG'
## Simple feature collection with 603 features and 11 fields (with 1 geometry empty)
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:   xmin: 477218.3 ymin: 75271.2 xmax: 833839.7 ymax: 295934.7
## Projected CRS: CH1903 / LV03

ws <- st_transform(ws, crs = 2056)

ws_freq_table <- subset(ws, ws$id %in% unique(ws_atlas$gis_id))

st_geometry(ws_freq_table) <- "geometry" #getting geometries for all sectors with WS INFO

grid.extent <- st_sf(a = 1:2, geom = st_sfc(st_point(x = c(5.959163471,
  45.816873303)), st_point(x = c(10.5765235389444, 47.8082225932288))),
  crs = 4326) %>%
  st_transform(2056)

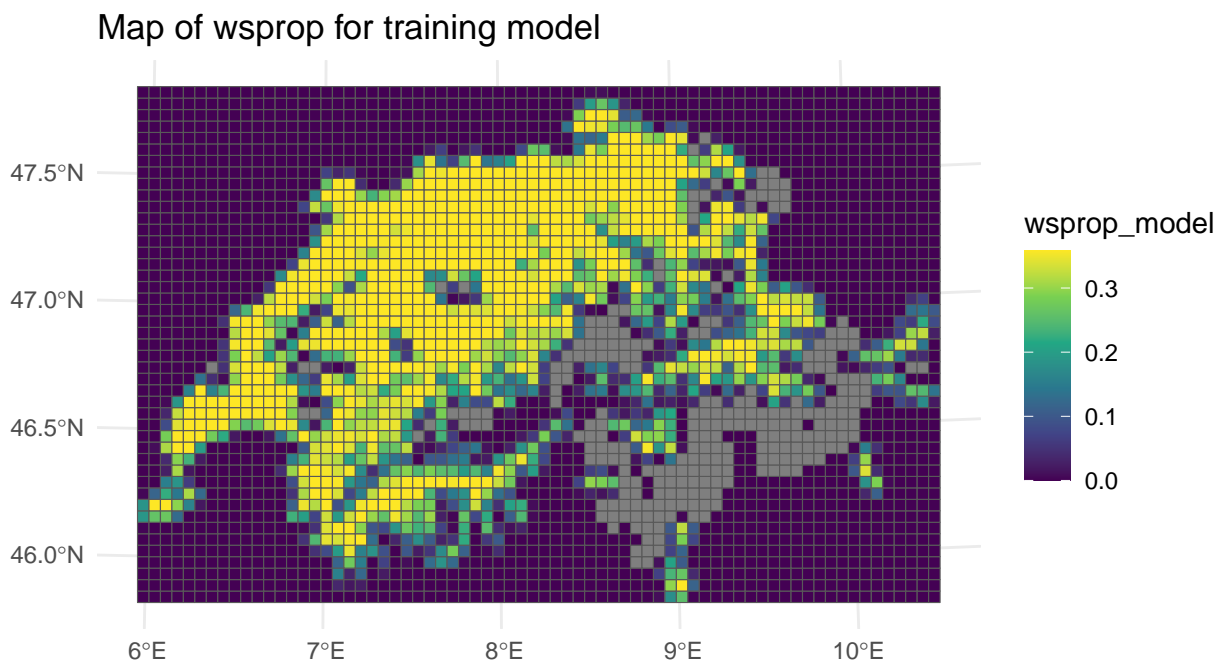
grid <- st_coordinates(grid.extent) %>%
  vect %>%
  rast(res = 5000, crs = "epsg:2056")
values(grid) <- ncell(grid):1 # create a 5000m grid that covers swiss extent

grid_ws_int <- terra::extract(grid, vect(ws_freq_table), method = "simple",
  touches = TRUE, exact = TRUE, xy = TRUE, ID = TRUE) #intersection ws
str(grid_ws_int)

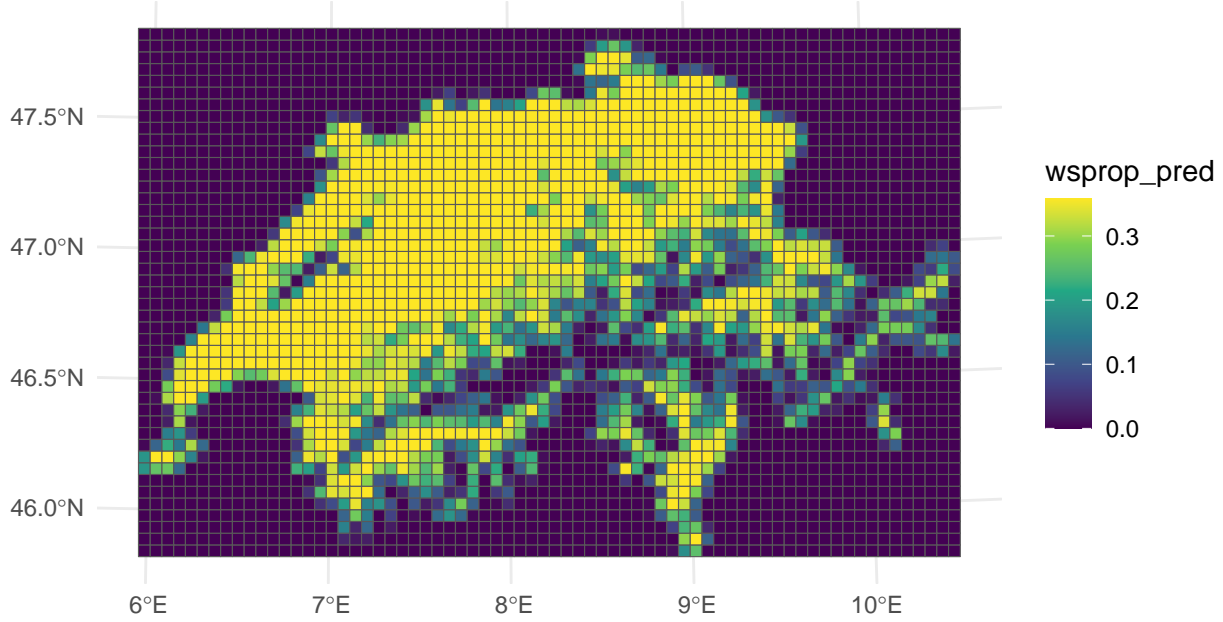
## 'data.frame':   3694 obs. of  5 variables:
##  $ ID      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ lyr.1    : int  791 790 722 721 720 652 651 650 582 581 ...
##  $ x        : num  2732500 2737500 2727500 2732500 2737500 ...
##  $ y        : num  1132500 1132500 1127500 1127500 1127500 ...
##  $ fraction: num  0.2039 0.1249 0.0839 0.7978 0.4634 ...
```

Nous avons dû prendre une certaine liberté quant à l'interprétation de ces fréquences, en voulant mettre en avant le point où une espèce est répandue au sein d'un secteur WS, et non nécessairement la quantité d'observations dans un secteur. Ce choix se montrera cohérent plus tard dans l'analyse, lorsque nous passerons à la binarisation de nos données pour la construction du modèle binomial.

Une distinction importante : En considérant le cas où nous n'avions pas d'observations dans la BD que nous pouvons spatialement relier à un secteur WS, nous injectons la médiane des fréquences pour chaque catégorie (fréquente (3) et rare (pas 3)), en fonction de la proportion de chaque carré couvert par le secteur WS en question. La fonction rend un tableau avec trois colonnes : "wsprop\_model", "wsprop\_pred" et les identifiants. La première ne contient pas ces injections. Nous verrons que le modèle n'est pas calibré sur les carrés pour lesquels nous manquons d'informations. En revanche, "wsprop\_pred" est la covariable que nous utiliserons pour faire les prédictions sur l'ensemble de la Suisse.



Map of wspread for model predictions



#### Explication de processus mathématiques importants:

Donc explicitement, nous modélisons la probability d'occupancy (chance de présence) pour de chaque carré en Suisse (je réprécise, qui sont cohérent entre l'atlas WS et la BD).

Mathématiquement cela s'exprime de la façon suivante:

$$z_{jt} \sim \text{Bernoulli}(\phi_{jt})$$

où  $z_{jt}$  et  $\phi_{jt}$  sont la variable réponse et la probability d'occupancy respectivement. Note:  $z_{jt}$  se réfère à la variable réponse au site  $j$  et au temps  $t$ .

Comme nous avons créer un modèle dynamique (temps), nous pouvons généraliser ce processus pour la totalité du modèle:

$$\sum_{t=1}^T \sum_{j=1}^J z_{jt} \sim \text{Binom}(TJ_{jt})$$

**Supposon importante:** Quelques suppositions importantes qui permette l'homogénéité du modèle.

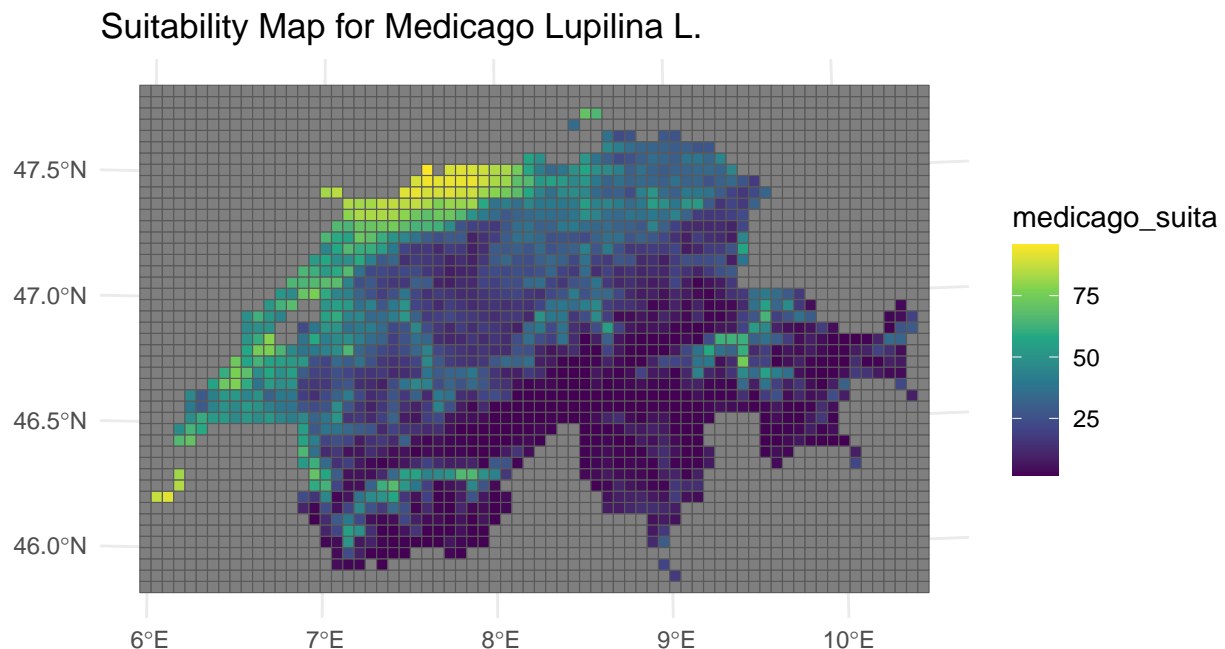
1. Aucune détection fausse positive
2. Toute variation dans l'occurrence est expliqué par les covariables (et non la qualité des données)
3. Les périodes des variables réponse sont indépendentes

**Les forumles du modèle:** Forumule pour Covariable d'Occurence et de Detection:

```
occ_formula <- ~scale(suitability) + I(scale(suitability)^2) +
  scale(wsprop) + I(scale(wsprop)^2)
det_formula <- ~scale(effort)
```

La formule d'occurrence prend en compte deux facteurs : l'adéquation de l'habitat (suitability) et la proportion d'occurrence dérivée de l'atlas de Welten et Sutter (wsprop). Note : nous standardisons et mettons les covariables sur le même ordre de grandeur pour obtenir une relation plus linéaire entre elles. Ces covariables ne sont pas dynamiques, et donc, dans un modèle à 30 pas de temps, ces données restent constantes. Cela est logique dans le cas de wsprop, car nos données sont limitées à 1994 (date du dernier supplément de l'Atlas WS). Dans le cas de l'adéquation de l'habitat (suitability), un modèle plus complet pourrait prendre en compte les changements environnementaux, mais cela dépasse le cadre de la présente étude.

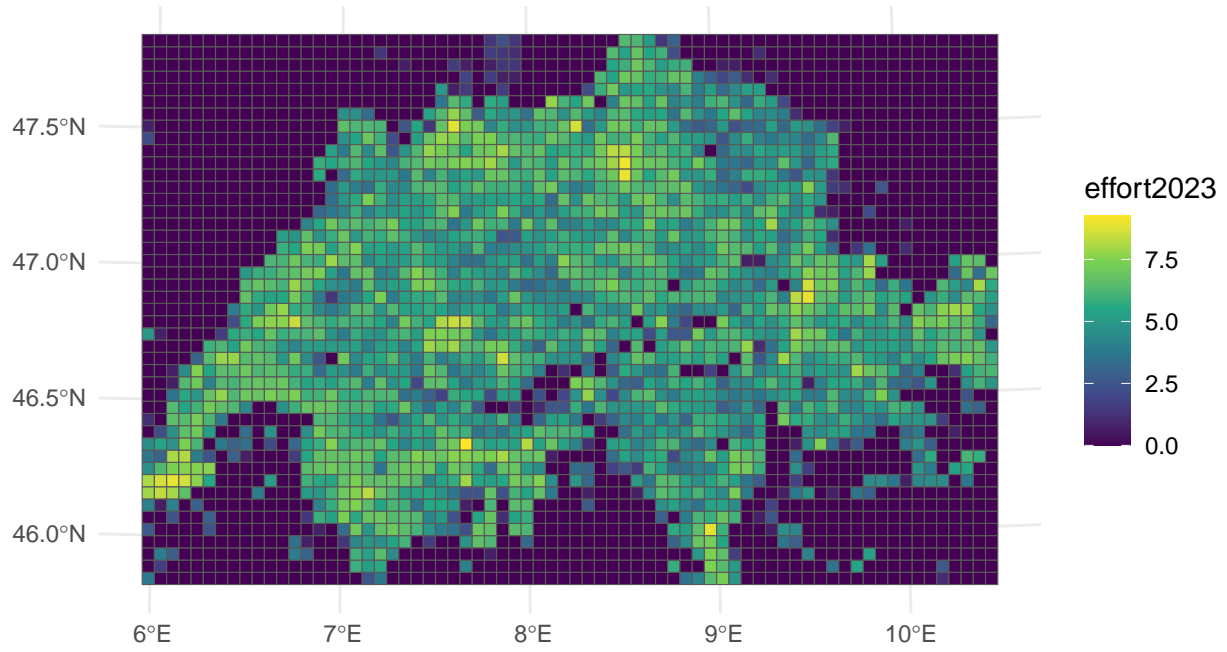
Vous trouverez ci-dessous le graphique de l'adéquation de l'habitat pour *Medicago lupulina* L.



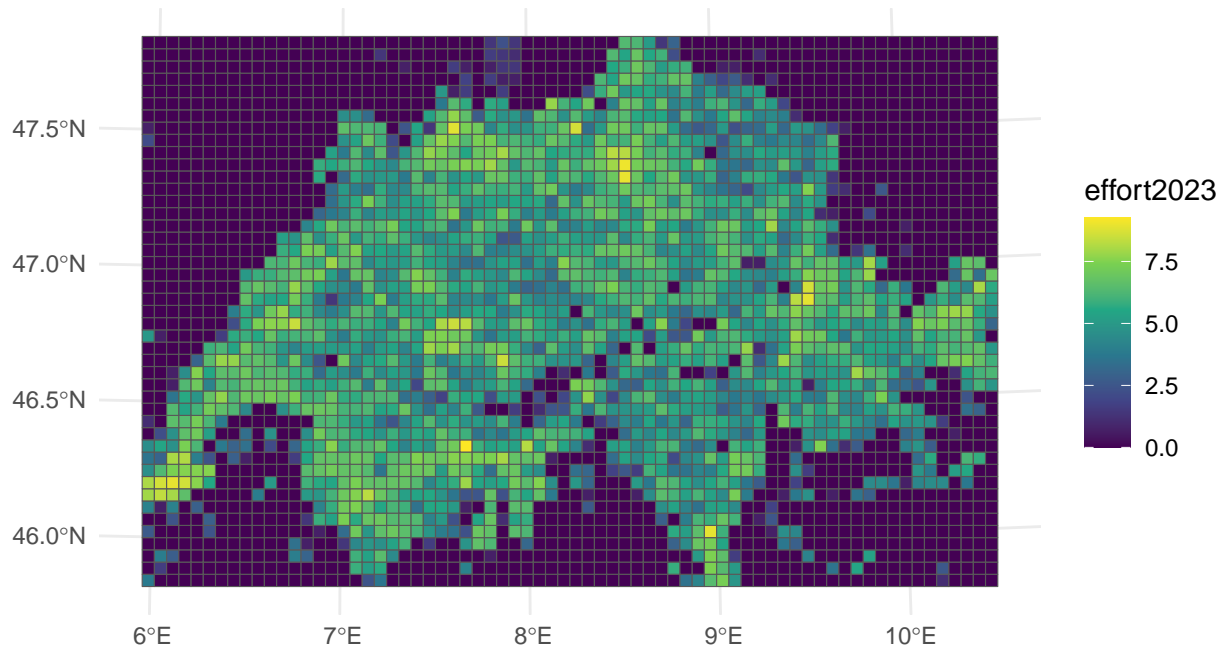
Dans notre modèle, l'effort d'échantillonnage a été calculé en prenant le logarithme de la somme de toutes les observations dans la base de données (BD), pour chaque carré de 5 km x 5 km et pour chacune des périodes de temps. Les résultats ont été enregistrés dans un fichier .Rdata qui est chargé dans le bloc de code ci-dessous. La covariable de détection a donc une dimension de 1849 x 30 (sites x périodes). Celle-ci n'est évidemment pas unique par taxon. Une visualisation de la covariable est présentée ci-dessous :



Effort map in 1994 (for all taxa)



Effort map in 2023 (for all taxa)



### Classification des covariables

Pourquoi ces facteurs ont-ils été classifiés comme covariables d'occurrence ou de détection ? Imaginons-nous au sein d'un groupe de dix personnes à la recherche de *Medicago lupulina*. La probabilité de présence, indépendamment du fait de la trouver ou non, dépend certainement de facteurs environnementaux (adéquation de l'habitat) ou bien des observations qui ont été faites dans le passé (*wsprop*). En revanche, la probabilité de trouver l'espèce dépendra aussi de la façon dont les dix membres du groupe se répartissent pour chercher dans le même carré ou dans plusieurs carrés. Ainsi, une covariable de détection n'a aucun impact sur la présence d'une espèce dans un carré, mais est directement liée aux variables qui influencent notre capacité à faire une observation.

### Le biais du modèle

La surface de la Suisse est couverte de 1849 carrés de  $25\text{km}^2$ . Dans le cas de *Medicago lupulina* L., le modèle est calibré uniquement sur 1267 ou 1603 carrés. Ces résultats sont obtenus après avoir filtré les carrés avec une valeur de *wsprop* égale à NA et, dans le cas du modèle réduit, en ayant également retiré les carrés dans lesquels nous n'avons aucune observation historique. Ce choix a été fait dans l'intention de réduire le biais de nos résultats. Cela dit, dans des cas comme ceux de *Saxifraga hirculus* et *Festuca valesiaca*, nous modélisons sur une surface notablement plus petite (26 et 105 carrés respectivement).

## Avantages et Desavantages pour modèle 'complet'

Avantage	Desavantage
Bien meilleur pour modèle à espèce rare	Plus long run-time, Biaisé

## Avantages et Desavantages pour modèle réduit

Avantage	Desavantage
Complexité plus petite, Moins biaisé	Très mauvaises prédictions si modèle est trop petit

**Construction du data pour le modèle:** Pour le choix du nombre de carrés sur lesquels le modèle sera entraîné, nous avons deux approches. Premièrement, nous pouvons simplement retirer le sous-ensemble de carrés présentant une contradiction entre la base de données (BD) et l'Atlas WS (246 carrés pour Medicago lupulina). Une deuxième approche consiste à prendre uniquement les carrés ayant historiquement au moins une observation sur les 30 périodes de temps.

```
medicago.eg_data <- readRDS("C:/Users/IF_240606/Desktop/Projet/DynamicModels30/Medicago_data.Rdata")
dim(medicago.eg_data) #1849 sites, 66 variables
```

```
## [1] 1849 66
```

```
medicago.eg_data_reduced <- medicago.eg_data %>%
  rowwise() %>%
  filter(sum(c_across(starts_with("presence"))) >= 1)
```

```
medicago.eg_data_reduced <- na.omit(medicago.eg_data_reduced) #NA wsprop sectors must be removed
dim(medicago.eg_data_reduced)
```

```
## [1] 1276 66
```

```
dim(na.omit(medicago.eg_data)) #sans filtre de obs dans secteurs
```

```
## [1] 1603 66
```

Le format final pour données du modèle:

```
## List of 4
## $ y : num [1:1276, 1:30, 1] 0 1 1 0 0 1 0 0 0 0 ...
## $ occ.covs:List of 4
## ..$ id_grid : int [1:1276] 46 47 48 88 89 117 118 119 159 160 ...
## ..$ wsprop : num [1:1276] 0.00891 0.17626 0.25407 0.03427 0.01686 ...
## ..$ suitability: num [1:1276] 0 0 0 0 0 ...
## ..$ years : num [1:1276, 1:30] 1994 1994 1994 1994 1994 ...
## $ det.covs:List of 3
## ..$ effort : num [1:1276, 1:30] 0 4.56 6.17 3.71 5.1 ...
## ..$ id_grid: int [1:1276] 46 47 48 88 89 117 118 119 159 160 ...
## ..$ years : num [1:1276, 1:30] 1994 1994 1994 1994 1994 ...
## $ coords : num [1:1276, 1:2] 2712500 2717500 2722500 2572500 2577500 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : NULL
## .. ..$ : chr [1:2] "X" "Y"
```

## Les parametres du modèle:

```
n.chains <- 3
n.batch <- 200
batch.length <- 25
n.samples <- n.batch * batch.length
ar1 <- TRUE
n.burn <- 2000
n.thin <- 12
```

1. `n.batch` : Fait référence au nombre d'échantillons Monte Carlo Markov Chain pour chaque chaîne, pour permettre le bon fonctionnement de notre échantillonneur MCMC.
2. `batch.length` : La longueur de chaque lot
3. `n.chains` : Nombre de chaînes en cours pour faire converger le modèle. Ce paramètre est en grande partie responsable de la complexité du modèle (~13 minutes pour Medicago L.).
4. `n.burn` : Les premiers N échantillons que l'on choisit d'ignorer, car l'intervalle de convergence est généralement trop large à ce stade.
5. `n.thin` : Un autre filtre pour les échantillons MCMC (après avoir éliminé `n.burn`). Dans notre modèle, nous sélectionnons chaque 12 résultat d'une chaîne de Markov.

Nous pouvons calculer manuellement le nombre d'échantillons du model au total, de la manière suivants:

$$\frac{(n.batch * batch.length) - n.burn}{n.thin} * n.chains = n.samples$$
$$\frac{(200 * 25) - 2000}{12} * 3 = 750$$

5. `AR1` : l'argument qui indique au modèle de prendre en compte l'autocorrélation temporelle. Le modèle ne serait pas dynamique. Si `AR1 = FALSE`, le modèle ne sera pas dynamique.

```
z.inits <- medicago.eg_data_list$y[, , 1]
inits <- list(beta = 0, alpha = 0, sigma.sq.psi = 1, z = z.inits)

priors <- list(beta.normal = list(mean = 0, var = 0.01517369),
  alpha.normal = list(mean = 0, var = 0.01517369), sigma.sq.psi.ig = list(a = 0.1,
    b = 0.1))
```

6. `inits` : Valeur initiales des parametres de l'algorithme MCMC. C'est valeur influence surtout la complexité du modèle et non les résultats. Dans nous modèle, nous avons utilisé les valeurs défaut.
7. `priors` : Une estimation a priori de la distribution des estimateurs.

Choix défaut pour la variable  $\sigma_{sq}$ , dépendance spatiale:

$$\sigma_{sq} \sim \text{InverseGamma}(\alpha, \beta)$$

Choix défaut pour la variable  $\phi$ , corrélation spatiale:

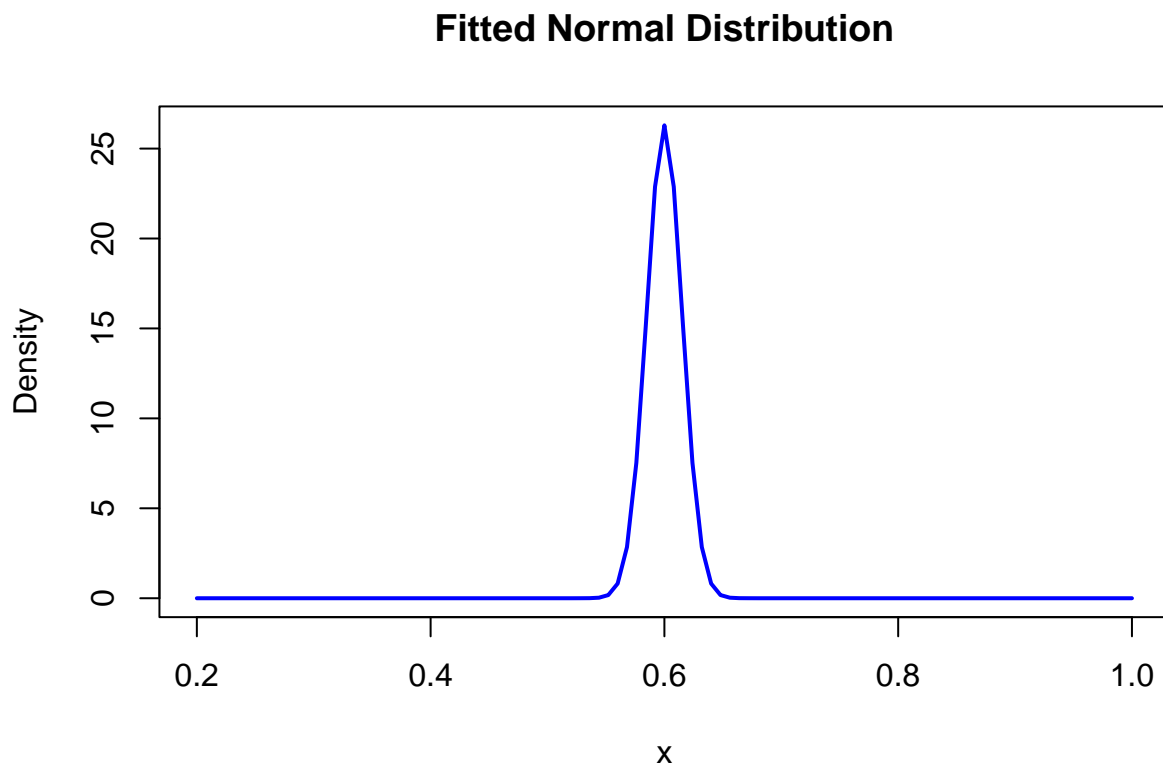
$$\phi \sim \text{Gamma}(\alpha, \beta)$$

Quand nous définissons nos priors, nous pouvons utiliser certains outils comme `brms` ou `rstan` pour appliquer nos contraintes dans la définition des priors. Premièrement, si nous savons que les valeurs désirées sont  $\in [0, \infty)$  ou  $\in [0, 1]$ , nos priors seront distribués en fonction de variables Gamma ou Beta respectives. Comme nos estimations sont à échelle logit elles peuvent prendre toutes valeurs entre  $(-\infty, \infty)$ , alors, nous voulons considéré la distribution Normale pour nos paramètres.

Supposons que nous souhaitions trouver les paramètres  $\mu$  et  $\sigma^2$ , d'une distribution Normale tel que la moyenne soit  $\mu = 0.6$  et 90% de la distribution se situe entre 0,4 et 0,8. Le code suivant nous permet de calculer  $\sigma^2$  en fonction du problème posé.

```
## Optimal mean (mu): 0.6

## Optimal standard deviation (sigma): 0.01517369
```



Ensuite, il faudrait tester la qualité du prior dans le modèle directement avec des données vides. Je n'ai pas pu faire le teste avec le package `spOccupancy`, mais je sais que c'est possible pour des modèles construits avec `brms` ou `rstan`.

D'après certaines sources, dans le cas d'un modèle à espèce unique, la définition précise des priors n'est pas essentielle. Ceci-dit, si dans l'avenir nous voudrions développer le modèle pour qu'il analyse les tendances parmi plusieurs espèces, il faudra être plus prudent en les définissant.

```
model.eg.medicago <- stPGOcc(occ.formula = occ_formula, det.formula = det_formula, data = medicago.eg_data_list, n.batch = n.batch, batch.length = batch.length, inits = inits_reduced, priors = priors, ar1 = ar1, n.burn = n.burn, n.thin = n.thin, n.chains = n.chains, n.report = 50, n.omp.threads = 11)
```

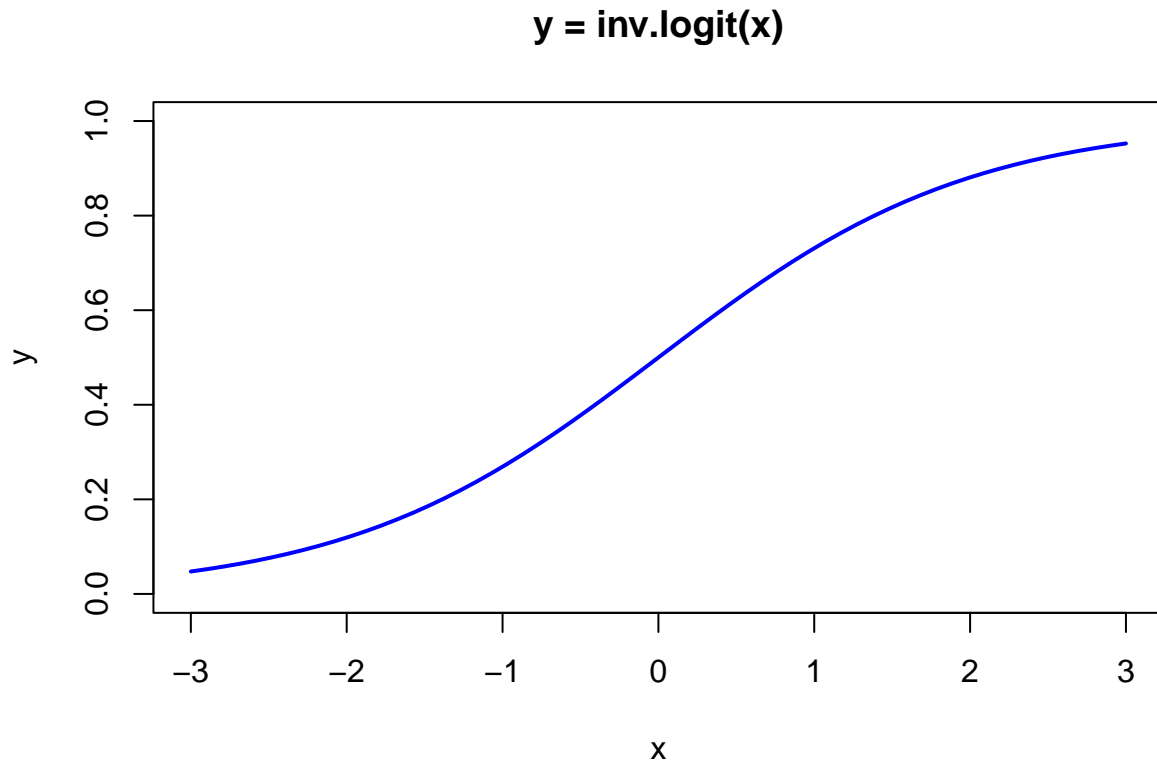
Dans l'intérêt de pouvoir générer le knitr du Rmarkdown, je ne passe pas le modèle dans ce markdown mais l'importe d'un résultat précédent. Noter que, si le seed n'est pas set, les résultats seront légèrement différents à cause du processus aléatoire des chaînes Markov.

```
model.eg.medicago <- readRDS("C:/Users/IF_240606/Desktop/Projet/DynamicModels30/MedicagoModel_reduced_Np")
summary(model.eg.medicago)
```

```
##
## Call:
## stPGOcc(occ.formula = occ_formula, det.formula = det_formula,
##   data = data_list_reduced, inits = inits_reduced, priors = priors,
##   n.batch = n.batch, batch.length = batch.length, n.omp.threads = 11,
##   ar1 = ar1, n.report = 50, n.burn = n.burn, n.thin = n.thin,
##   n.chains = n.chains)
##
## Samples per Chain: 5000
## Burn-in: 2000
## Thinning Rate: 12
## Number of Chains: 3
## Total Posterior Samples: 750
## Run Time (min): 9.616
##
## Occurrence (logit scale):
##           Mean      SD   2.5%   50%   97.5%   Rhat ESS
## (Intercept) -0.1272 0.1118 -0.3482 -0.1252  0.0821 1.2327  51
## scale(suitability) 0.3583 0.0440  0.2703  0.3588  0.4426 1.0318 390
## I(scale(suitability)^2) 0.0479 0.0341 -0.0162  0.0484  0.1134 1.0093 223
## scale(wsprop) 0.2246 0.0472  0.1370  0.2255  0.3179 1.0509 395
## I(scale(wsprop)^2) -0.0781 0.0316 -0.1456 -0.0768 -0.0199 1.0786 247
##
## Detection (logit scale):
##           Mean      SD   2.5%   50%   97.5%   Rhat ESS
## (Intercept)  0.6047 0.0609 0.4926 0.6039 0.7148 1.0673 217
## scale(effort) 0.6579 0.0352 0.5879 0.6577 0.7233 0.9997 579
##
## Spatio-temporal Covariance:
##           Mean      SD   2.5%   50%   97.5%   Rhat ESS
## sigma.sq  0.6932 0.1001 0.5295 0.6799 0.8984 1.0006 183
## phi       0.0001 0.0000 0.0000 0.0001 0.0001 1.0218  94
## sigma.sq.t 1.0071 0.2873 0.5745 0.9629 1.6550 1.0198 445
## rho       0.5372 0.0972 0.3351 0.5394 0.7035 1.0335 197
```

**Analyse des résultats du modèle:** L'échelle logit:

```
x <- seq(-3, 3, length.out = 60)
y <- inv.logit(x)
plot(x, y, type = "l", col = "blue", lwd = 2, main = "y = inv.logit(x)",
      xlab = "x", ylab = "y", ylim = c(0, 1))
```



Nous utilisons la translation principalement car elle permet d'établir une relation linéaire entre les covariables:

$$\text{logit}(p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots$$

ESS - Effective Sample Size.

Rhat - La qualité des estimateurs sont généralement très bon. Nous cherchons à ce qu'il soit  $< 1.1$ . C'est donc le cas de tous sauf 'Intercept' et 'phi'

sigma.sq - La variable qui représente la force de la dépendance spatiale.

phi - Variable qui contrôle la corrélation spatiale. Dans notre modèle, la corrélation spatiale diminue très rapidement avec de la distance.

sigma.sq.t - La dépendance temporelle.

rho: un paramètre qui indique la probabilité conditionnelle que site A à temps 2 soit occupé si site A à temps 1 l'a été aussi. Comme  $\rho = 0.537 > 0$ , si le site est occupé au T1, il aura plus de chance d'être occupé au T2. Si  $\rho < 0$ , le modèle insinue qu'une présence T1 suggère une absence T2.

**Prédiction du modèle sur extent Suisse:** Nous standardisons les covariables d'occupancy (suitability et wsprop (prédiction)) et créons l'objet de prediction qui contient les informations injecté à chaque site.

```
# Creation de l'objet X.0 du prédiction: Number of
# prediction sites.
J.pred <- 1849
# Number on.sites# Number of prediction years.
n.years.pred <- 30
# Number of predictors (including intercept)
p.occ <- ncol(model.eg.medicago$beta.samples)
suitability.pred <- (medicago.eg_data$suitability - mean(medicago.eg_data$suitability))/sd(medicago.eg_data$suitability)
wsprop.pred <- medicago.eg_data$wsprop_pred - mean(medicago.eg_data$wsprop_pred)/sd(medicago.eg_data$wsprop_pred)
X.0 <- array(1, dim = c(J.pred, n.years.pred, p.occ)) # Create three-dimensional array
X.0[, , 2] <- suitability.pred # suitability
X.0[, , 3] <- suitability.pred^2 # suitability^2
X.0[, , 4] <- wsprop.pred # wsprop
X.0[, , 5] <- wsprop.pred^2 # wsprop^2
str(X.0) #structure
```

```
## num [1:1849, 1:30, 1:5] 1 1 1 1 1 1 1 1 1 1 ...
```

### Fonction Prédiction:

```
## Warning: st_centroid assumes attributes are constant over geometries
```

La structure des résultats de la fonction 'prédicit':

```
## List of 6
## $ z.0.samples : num [1:750, 1:1849, 1:30] 0 0 0 0 0 0 0 0 1 0 ...
## $ w.0.samples : 'mcmc' num [1:750, 1:1849] -1.179 -0.603 0.369 -1.029 -0.553 ...
## ..- attr(*, "mcpars")= num [1:3] 1 750 1
## $ psi.0.samples: num [1:750, 1:1849, 1:30] 0.0982 0.1886 0.3503 0.1124 0.2185 ...
## $ run.time : 'proc_time' Named num [1:5] 9.37 0.48 10.58 NA NA
## ..- attr(*, "names")= chr [1:5] "user.self" "sys.self" "elapsed" "user.child" ...
## $ call : language predict.stPGOcc(object = model.eg.medicago, X.0 = X.0, coords.0 = coords.0)
## $ object.class : chr "stPGOcc"
## - attr(*, "class")= chr "predict.stPGOcc"
```

### Explications des variables résultantes:

La première dimension des résultats indique le nombre total d'échantillons MCMC, comme nous l'avons discuté précédemment.

z.0.samples : La prédiction de présence (0/1) pour chacun des 1849 sites pour chaque année.

w.0.samples : Les prédictions a posteriori des échantillons qui résument la dépendance spatiale par site.

psi.0.samples : La prédiction d'occupation [0,1] pour chacun des 1849 sites pour chaque année.

**Cartes pour visualisation des résultats par espèce** Nous graphons uniquement les cartes pour les années d'extrémités



```

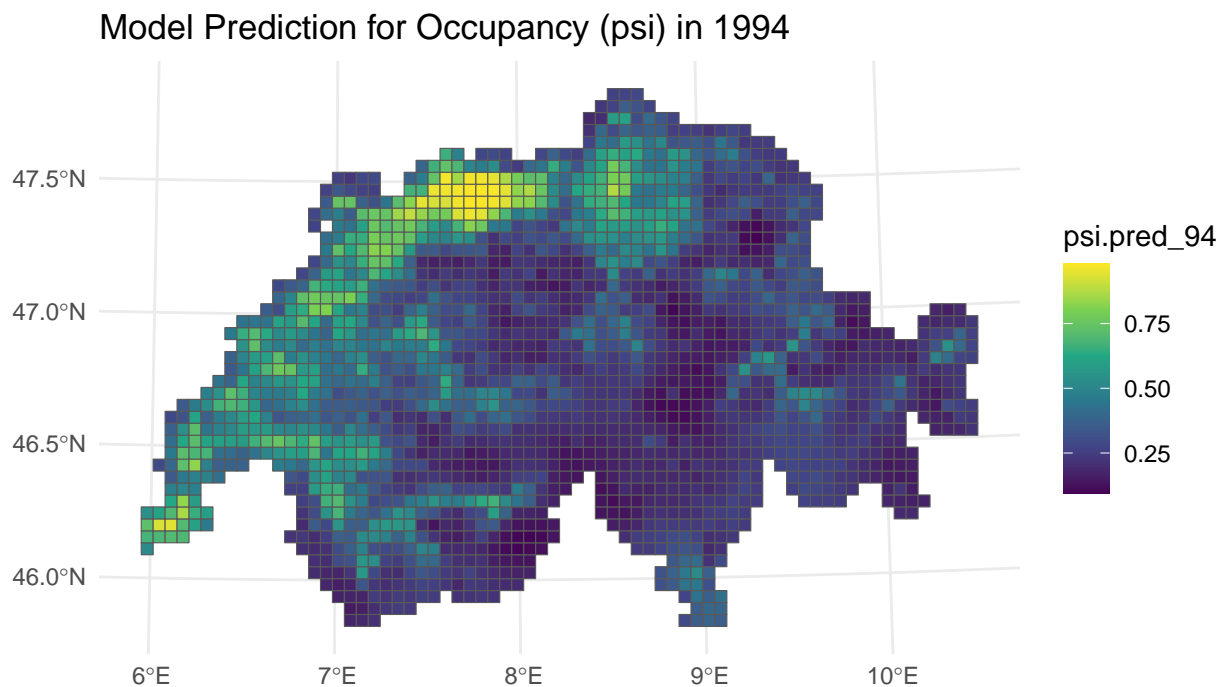
mean.1994.psi = apply(medicago.eg_pred$psi.0.samples[, , 1],
  2, mean) #[, , 1] se réfère à la première année
mean.2023.psi = apply(medicago.eg_pred$psi.0.samples[, , 30],
  2, mean) #année 2023
medicago.eg_data$psi.pred_94 <- mean.1994.psi
medicago.eg_data$psi.pred_23 <- mean.2023.psi

# plot simultané
gr.psi.pred_94 <- ggplot(medicago.eg_data) + geom_sf(aes(fill = psi.pred_94)) +
  scale_fill_viridis_c() + labs(title = ("Model Prediction for Occupancy (psi) in 1994")) +
  theme_minimal()

gr.psi.pred_23 <- ggplot(medicago.eg_data) + geom_sf(aes(fill = psi.pred_23)) +
  scale_fill_viridis_c() + labs(title = ("Model Prediction for Occupancy (psi) in 2023")) +
  theme_minimal()

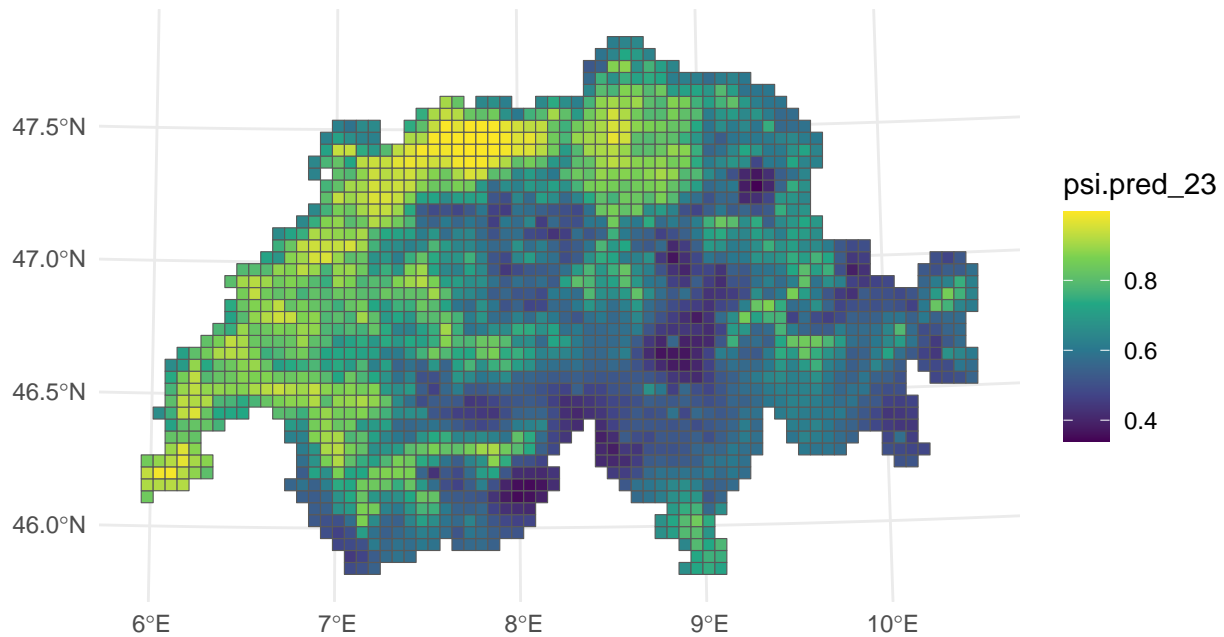
gr.psi.pred_94

```



```
gr.psi.pred_23
```

## Model Prediction for Occupancy (psi) in 2023



Graphes comparants les résultats du modèle avec la BD Prenons la somme des z.0.samples pour chaque année puis comparons le résultat avec la DB binarisé en présences/absences.

```
years_red <- c(1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,
  2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011,
  2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
  2022, 2023) #for plotting

# sum to find predicted presences (z-samples)
total_presence <- vector("numeric", length = 30)
for (i in 1:30) {
  total_presence[i] <- sum(st_drop_geometry(medicago.eg_data[,
    i + 34]))
}

sum_psi_sites_med <- apply(medicago.eg_pred$psi.0.samples, c(1,
  3), sum)
psi_medicago <- colMeans(sum_psi_sites_med)
sd_psi_medicago <- apply(sum_psi_sites_med, 2, sd)

# mean zsamples across all mcmc samples
z_0_sum_all_times <- apply(medicago.eg_pred$z.0.samples, c(1,
  3), sum)
mean_each_time <- apply(z_0_sum_all_times, 2, mean)
sd_zsample_medicago <- apply(z_0_sum_all_times, 2, sd)

sum_zsamples_reduced <- data.frame(Time = years_red, sum_zsamples = mean_each_time,
```

```

presence = total_presence, psi_med = psi_medicago, sd_psi = sd_psi_medicago,
sd_zsample = sd_zsample_medicago)

ggplot(sum_zsamples_reduced[2:30, ], aes(x = Time)) + geom_line(aes(y = sum_zsamples,
  color = "Pred Detections (Zsample)")) + geom_point(aes(y = sum_zsamples,
  color = "Pred Detections (Zsample)", shape = "Pred Detections (Zsample)")) +
geom_line(aes(y = presence, color = "Presence")) + geom_point(aes(y = presence,
  color = "Presence", shape = "Presence")) + geom_line(aes(y = psi_med,
  color = "Occupancy Prediction (psi)")) + geom_point(aes(y = psi_med,
  color = "Occupancy Prediction (psi)", shape = "Occupancy Prediction (psi)")) +
# geom_errorbar(aes(y = sum_zsamples, ymin = sum_zsamples -
# sd_zsample, ymax = sum_zsamples + sd_zsample), color =
# 'blue', width = 0.2) + geom_errorbar(aes(y = psi_med,
# ymin = psi_med - sd_psi, ymax = psi_med + sd_psi), color
# = 'green', width = 0.2) +

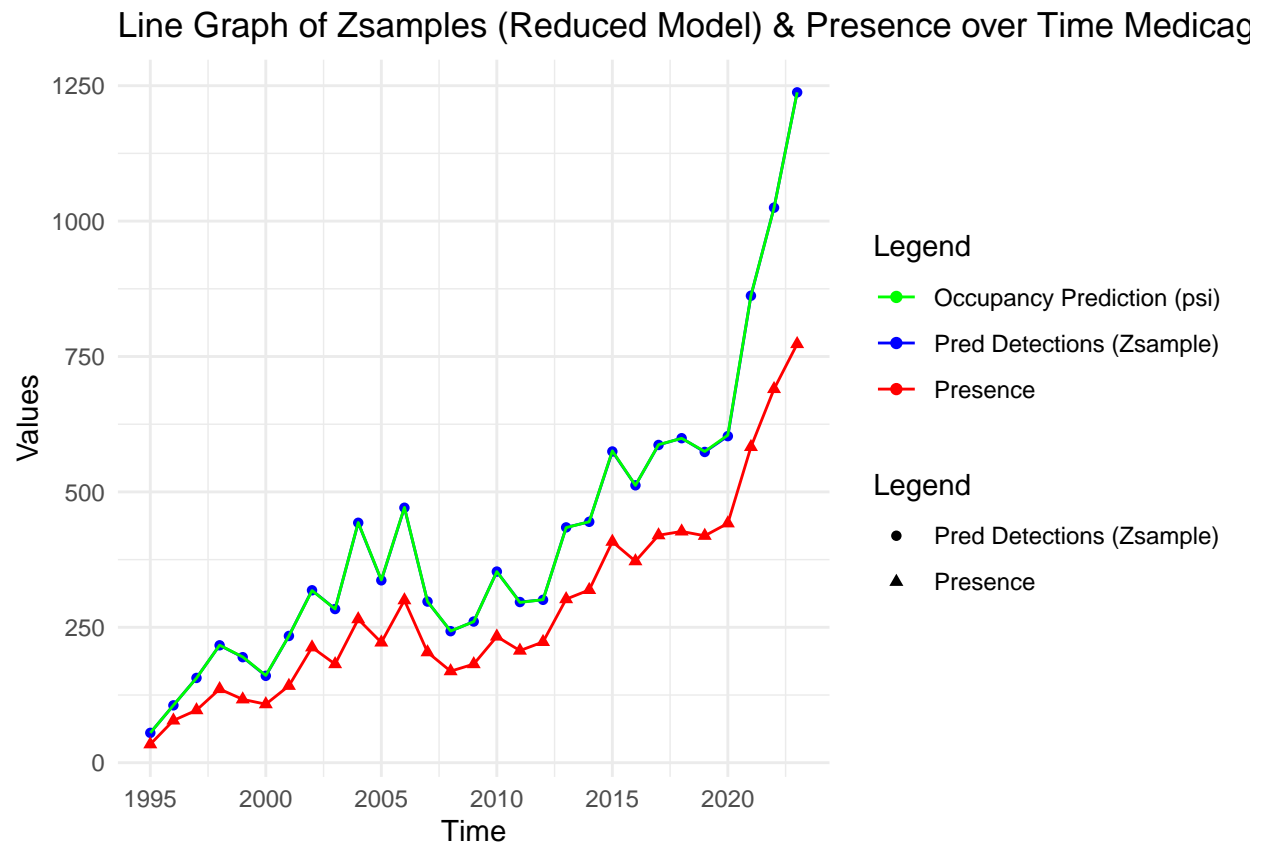
labs(x = "Time", y = "Values", title = ("Line Graph of Zsamples (Reduced Model) & Presence over Time Medicag
  scale_color_manual(values = c(`Pred Detections (Zsample)` = "blue",
  Presence = "red", `Occupancy Prediction (psi)` = "green"),
  name = "Legend") + scale_shape_manual(values = c(`Pred Detections (Zsample)` = 16,
  Presence = 17, `Occupancy Prediction (psi)` = 15), name = "Legend") +
theme_minimal()

```

```

## Warning: Removed 29 rows containing missing values or values outside the scale range
## (`geom_point()`).

```



Note: nous graphons uniquement d'années 1995-2023.

Répetons ce processus pour les deux autres taxon. Dans l'intérêt de brièveté, je vous épargne le code en complet et vais simplement utiliser la fonction “readRDS” avec les données que j'ai déjà sauvegardé.

Modèle sur tous les sites saufs ceux qui n'ont pas d'observations dans leur WS respectif

```
medicago_full.pred <- readRDS(paste0("C:/Users/IF_240606/Desktop/Projet/",  
  "DynamicModels30/stPGOcc_Medicago_outpred.Rdata"))  
str(medicago_full.pred)
```

```
## List of 6  
## $ z.0.samples : num [1:750, 1:1849, 1:30] 0 0 0 0 0 0 0 0 0 0 ...  
## $ w.0.samples : 'mcmc' num [1:750, 1:1849] -0.963 -1.442 -1.951 -0.925 -1.025 ...  
## ..- attr(*, "mcpar")= num [1:3] 1 750 1  
## $ psi.0.samples: num [1:750, 1:1849, 1:30] 0.0273 0.0172 0.0116 0.0302 0.0255 ...  
## $ run.time : 'proc_time' Named num [1:5] 7.56 0.33 8.5 NA NA  
## ..- attr(*, "names")= chr [1:5] "user.self" "sys.self" "elapsed" "user.child" ...  
## $ call : language predict.stPGOcc(object = out, X.0 = X.0, coords.0 = coords.full, t.cols =  
## $ object.class : chr "stPGOcc"  
## - attr(*, "class")= chr "predict.stPGOcc"
```

```
medicago_spi <- read.csv("C:/Users/IF_240606/Desktop/Projet/SPIcomp/1028530_Medicago_lupulina_proba_mat.  
  
# sum to find predicted presences (z-samples)  
total_presence <- vector("numeric", length = 30)  
for (i in 1:30) {  
  total_presence[i] <- sum(st_drop_geometry(medicago_eg_data[,  
    i + 34]))  
}  
sum_psi_sites_med <- apply(medicago_full.pred$psi.0.samples,  
  c(1, 3), sum)  
psi_medicago <- colMeans(sum_psi_sites_med)  
sd_psi_medicago <- apply(sum_psi_sites_med, 2, sd)  
  
# data for spi approche  
medicago_spi_yr_sum <- as.data.frame(colSums(medicago_spi, na.rm = T)[(ncol(medicago_spi) -  
  28):ncol(medicago_spi)])  
colnames(medicago_spi_yr_sum) <- c("spi_sum")  
medicago_spi_yr_sum <- rbind(medicago_spi_yr_sum, NA)  
  
# mean zsamples across all mcmc samples  
z_0_sum_all_times <- apply(medicago_full.pred$z.0.samples, c(1,  
  3), sum)  
mean_each_time <- apply(z_0_sum_all_times, 2, mean)  
sd_zsample_medicago <- apply(z_0_sum_all_times, 2, sd)  
  
sum_zsamples_reduced <- data.frame(Time = years_red, sum_zsamples = mean_each_time,  
  presence = total_presence, psi_med = psi_medicago, sd_psi = sd_psi_medicago,  
  sd_zsample = sd_zsample_medicago, spi_sum = medicago_spi_yr_sum)  
  
ggplot(sum_zsamples_reduced[2:30, ], aes(x = Time)) + geom_line(aes(y = sum_zsamples,  
  color = "Pred Detections (Zsample)")) + geom_point(aes(y = sum_zsamples,  
  color = "Pred Detections (Zsample)", shape = "Pred Detections (Zsample)")) +  
geom_line(aes(y = presence, color = "Presence")) + geom_point(aes(y = presence,  
  color = "Presence", shape = "Presence")) + geom_line(aes(y = psi_med,  
  color = "Occupancy Prediction (psi)")) + geom_point(aes(y = psi_med,  
  color = "Occupancy Prediction (psi)", shape = "Occupancy Prediction (psi)")) +
```

```

geom_line(aes(y = spi_sum, color = "SPI results Medicago")) +
  geom_point(aes(y = spi_sum, color = "SPI results Medicago")) +
  # geom_errorbar(aes(y = sum_zsamples, ymin = sum_zsamples -
  # sd_zsample, ymax = sum_zsamples + sd_zsample), color =
  # 'blue', width = 0.2)+

  # geom_errorbar(aes(y = psi_med, ymin = psi_med - sd_psi,
  # ymax = psi_med + sd_psi), color = 'green', width = 0.2)+

labs(x = "Time", y = "Values", title = ("Line Graph of Zsamples (Reduced Model) & Presence over Time Medicago")) +
scale_color_manual(values = c(`Pred Detections (Zsample)` = "blue",
  Presence = "red", `Occupancy Prediction (psi)` = "green"),
  name = "Legend") + scale_shape_manual(values = c(`Pred Detections (Zsample)` = 16,
  Presence = 17, `Occupancy Prediction (psi)` = 15), name = "Legend") +
theme_minimal()

```

```

## Warning: Removed 29 rows containing missing values or values outside the scale range
## (`geom_point()`).

```

```

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

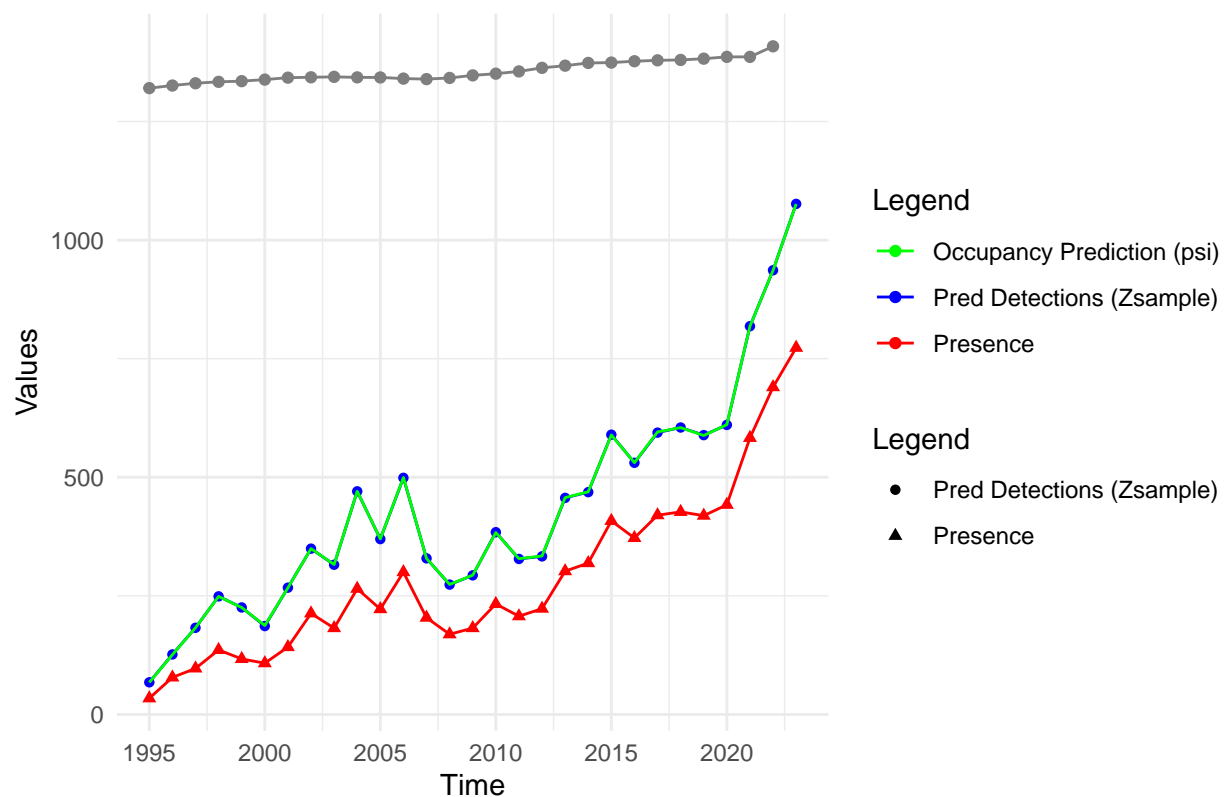
```

```

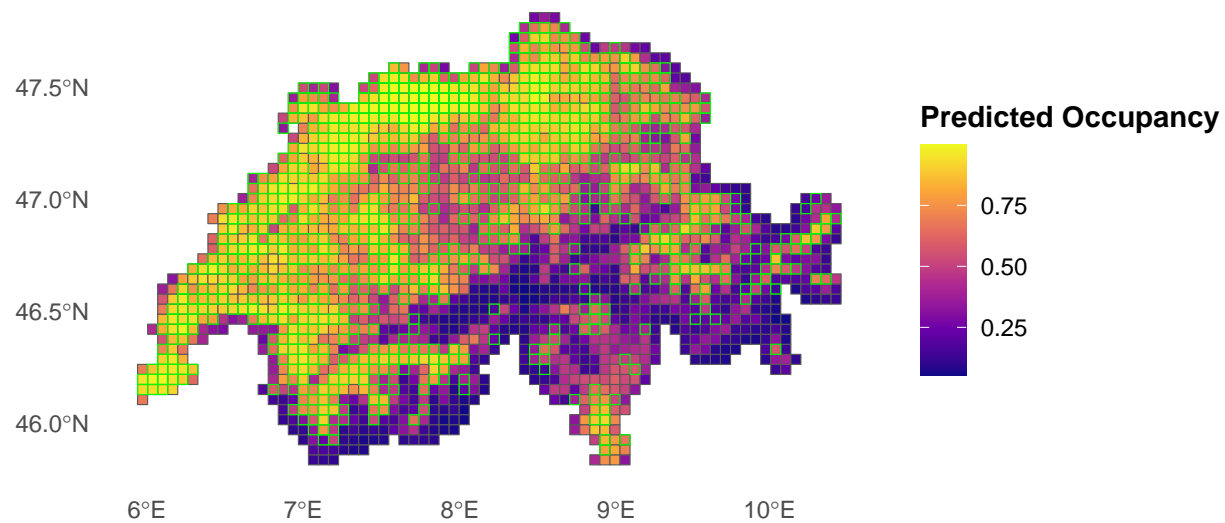
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).

```

Line Graph of Zsamples (Reduced Model) & Presence over Time Medicago



## Medicago Lupulina I Response Variable with Model Predicted Occ



## Festuca Valesiaca Gaudin:

```
str(festuca.eg_data_list_red)
```

```
## List of 4
## $ y      : num [1:105, 1:30, 1] 0 1 1 1 0 1 1 0 1 1 ...
## $ occ.covs:List of 4
## ..$ id_grid : int [1:105] 20 168 229 230 238 298 299 304 308 309 ...
## ..$ wsprop   : num [1:105] 0 0 0.461 0.255 0.156 ...
## ..$ suitability: num [1:105] 0 0 24.1 17.5 14.1 ...
## ..$ years    : num [1:105, 1:30] 1994 1994 1994 1994 1994 ...
## $ det.covs:List of 3
## ..$ effort : num [1:105, 1:30] 2.2 3.61 5.89 5.66 7.45 ...
## ..$ id_grid: int [1:105] 20 168 229 230 238 298 299 304 308 309 ...
## ..$ years  : num [1:105, 1:30] 1994 1994 1994 1994 1994 ...
## $ coords  : num [1:105, 1:2] 2582500 2622500 2577500 2582500 2622500 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : NULL
## .. ..$ : chr [1:2] "X" "Y"
```

```
summary(model.eg.festuca)
```

```
##
## Call:
## stPGOcc(occ.formula = occ_formula, det.formula = det_formula,
## data = data_list_reduced, inits = inits_reduced, priors = priors,
## n.batch = n.batch, batch.length = batch.length, n.omp.threads = 11,
## ar1 = ar1, n.report = 50, n.burn = n.burn, n.thin = n.thin,
## n.chains = n.chains)
##
## Samples per Chain: 5000
## Burn-in: 2000
## Thinning Rate: 12
## Number of Chains: 3
## Total Posterior Samples: 750
## Run Time (min): 0.8538
##
## Occurrence (logit scale):
##           Mean      SD    2.5%    50%  97.5%   Rhat ESS
## (Intercept)   -0.0643 0.1267 -0.3225 -0.0645 0.1695 1.0112 368
## scale(suitability) 0.2652 0.0911 0.0832 0.2666 0.4404 1.0089 750
## I(scale(suitability)^2) 0.1227 0.0719 -0.0226 0.1219 0.2575 1.0105 645
## scale(wsprop) 0.3406 0.0997 0.1463 0.3381 0.5257 1.0271 608
## I(scale(wsprop)^2) -0.0289 0.1064 -0.2466 -0.0311 0.1728 1.0124 530
##
## Detection (logit scale):
##           Mean      SD    2.5%    50%  97.5%   Rhat ESS
## (Intercept) 0.1290 0.1166 -0.0916 0.1277 0.3598 1.0077 456
## scale(effort) 0.3978 0.0944 0.2236 0.3940 0.5781 1.0088 750
##
## Spatio-temporal Covariance:
##           Mean      SD    2.5%    50%  97.5%   Rhat ESS
## sigma.sq 0.4785 0.1883 0.1954 0.4512 0.9180 1.0569 241
```



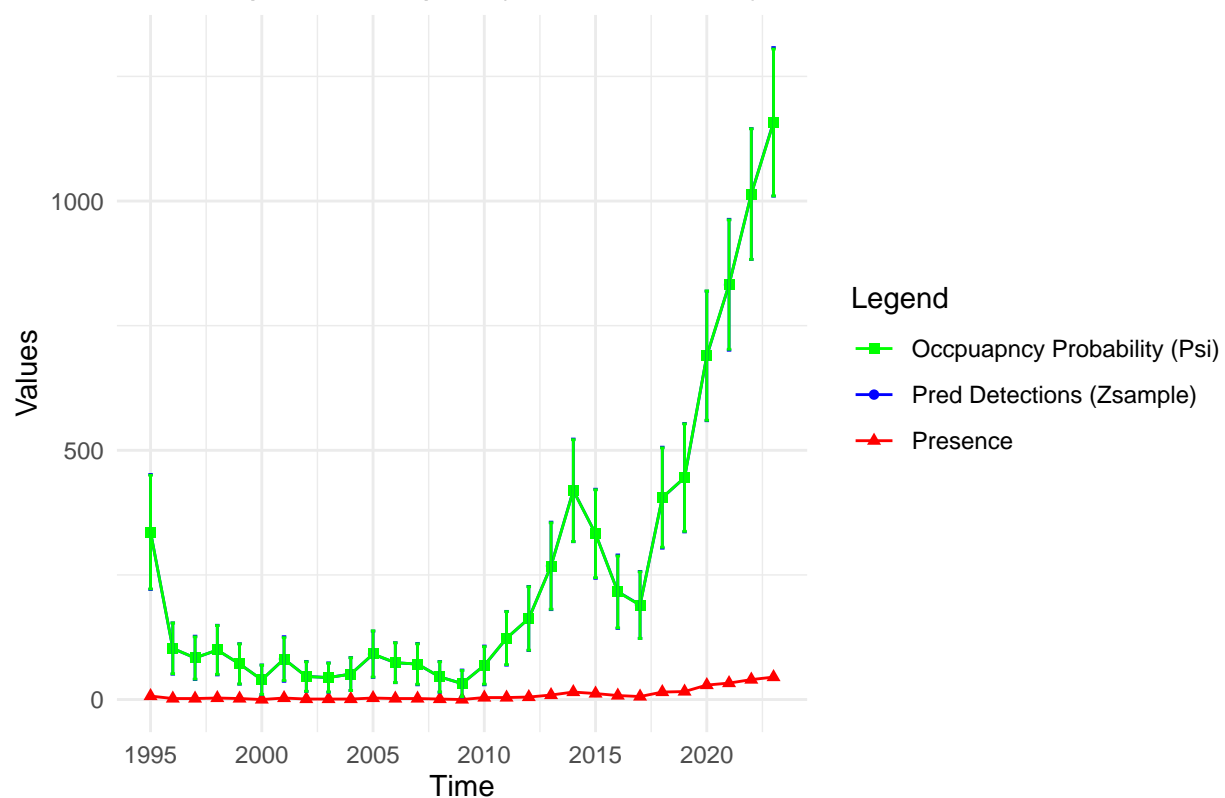
```
## phi          0.0003 0.0001 0.0001 0.0003 0.0006 1.0031 330
## sigma.sq.t   3.7875 1.3469 1.9633 3.5896 6.9011 1.0267 417
## rho          0.6750 0.0786 0.5003 0.6829 0.8048 1.0022 588
```

```
str(model.pred.eg.festuca)
```

```
## List of 6
## $ z.0.samples : num [1:750, 1:1849, 1:30] 1 1 1 0 1 1 1 1 1 1 ...
## $ w.0.samples : 'mcmc' num [1:750, 1:1849] -0.578 -0.553 0.645 0.451 -0.27 ...
## ..- attr(*, "mcpair")= num [1:3] 1 750 1
## $ psi.0.samples: num [1:750, 1:1849, 1:30] 0.809 0.556 0.814 0.787 0.795 ...
## $ run.time      : 'proc_time' Named num [1:5] 14.55 0.49 15.91 NA NA
## ..- attr(*, "names")= chr [1:5] "user.self" "sys.self" "elapsed" "user.child" ...
## $ call          : language predict.stPGOcc(object = out_reduced, X.0 = X.0, coords.0 = coords.full,
## $ object.class  : chr "stPGOcc"
## - attr(*, "class")= chr "predict.stPGOcc"
```

Le modèle est calibré sur 105 sites/1849.

Line Graph of Zsamples (Reduced Model) & Presence over Time Festuca



## Modèle “full” pour Festuca Valesiaca Gaudin.:

```
model.eg.festuca_full <- readRDS(paste0("C:/Users/IF_240606/Desktop/Projet/",
  "DynamicModels30/stPGOcc.Festuca.Model.Rdata"))
summary(model.eg.festuca_full)
```

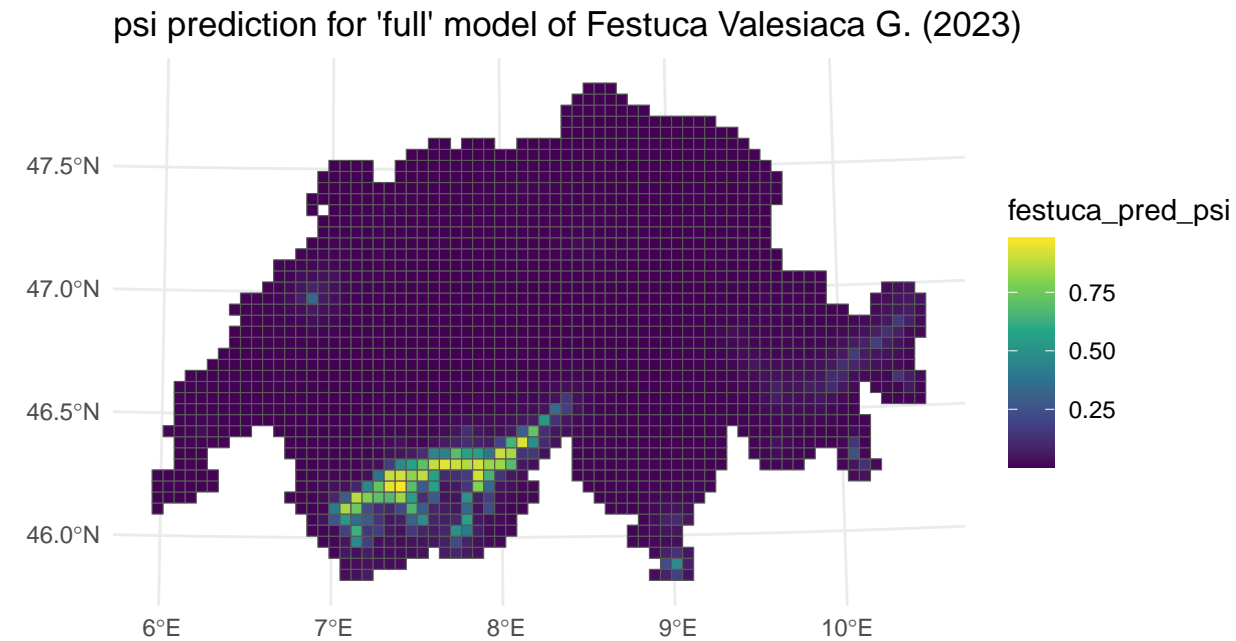
```
##
## Call:
## stPGOcc(occ.formula = occ_formula, det.formula = det_formula,
##   data = data_list, inits = inits, priors = priors, n.batch = n.batch,
##   batch.length = batch.length, n.omp.threads = 11, ar1 = ar1,
##   n.report = 50, n.burn = n.burn, n.thin = n.thin, n.chains = n.chains)
##
## Samples per Chain: 5000
## Burn-in: 2000
## Thinning Rate: 12
## Number of Chains: 3
## Total Posterior Samples: 750
## Run Time (min): 13.4143
##
## Occurrence (logit scale):
##           Mean      SD   2.5%   50%  97.5%   Rhat ESS
## (Intercept)   -0.0708 0.1153 -0.2830 -0.0738 0.1639 1.0046 142
## scale(suitability) 0.2882 0.1089 0.0863 0.2882 0.4987 1.1432 158
## I(scale(suitability)^2) -0.0317 0.0473 -0.1234 -0.0304 0.0628 1.0592 228
## scale(wsprop) 0.2401 0.1143 -0.0016 0.2409 0.4688 1.0221 352
## I(scale(wsprop)^2) 0.0852 0.0271 0.0302 0.0849 0.1387 1.1962 116
##
## Detection (logit scale):
##           Mean      SD   2.5%   50%  97.5%   Rhat ESS
## (Intercept) 0.1126 0.1134 -0.1216 0.1178 0.3330 1.0060 489
## scale(effort) 0.4576 0.0908 0.2851 0.4567 0.6466 1.0377 414
##
## Spatio-temporal Covariance:
##           Mean      SD   2.5%   50%  97.5%   Rhat ESS
## sigma.sq 9.9683 4.0912 4.3433 9.2923 19.8947 2.9446 23
## phi      0.0000 0.0000 0.0000 0.0000 0.0001 8.9734 5
## sigma.sq.t 9.3277 3.2049 4.7288 8.7712 16.9655 1.2823 96
## rho      0.8569 0.0403 0.7631 0.8617 0.9203 1.4911 55
```

```
festuca_full.pred <- readRDS(paste0("C:/Users/IF_240606/Desktop/Projet/",
  "DynamicModels30/stPGOcc_Festuca_outpred.Rdata"))
str(festuca_full.pred)
```

```
## List of 6
## $ z.0.samples : num [1:750, 1:1849, 1:30] 0 0 0 0 0 0 0 0 0 0 ...
## $ w.0.samples : 'mcmc' num [1:750, 1:1849] -0.1282 -0.5464 -0.0846 -0.6616 -1.3152 ...
## ..- attr(*, "mcp")= num [1:3] 1 750 1
## $ psi.0.samples: num [1:750, 1:1849, 1:30] 0.0207 0.0177 0.0276 0.0144 0.0121 ...
## $ run.time : 'proc_time' Named num [1:5] 7.19 0.37 8.08 NA NA
## ..- attr(*, "names")= chr [1:5] "user.self" "sys.self" "elapsed" "user.child" ...
## $ call : language predict.stPGOcc(object = out, X.0 = X.0, coords.0 = coords.full, t.cols =
## $ object.class : chr "stPGOcc"
## - attr(*, "class")= chr "predict.stPGOcc"
```

```
festuca.eg_data$festuca_pred_psi <- apply(festuca_full.pred$psi.0.samples[,
, 30], 2, mean)

ggplot(festuca.eg_data) + geom_sf(aes(fill = festuca_pred_psi)) +
  scale_fill_viridis_c() + labs(title = "psi prediction for 'full' model of Festuca Valesiaca G. (2023)")
  theme_minimal()
```



```
festuca_spi <- read.csv("C:/Users/IF_240606/Desktop/Projet/SPIcomp/1019640_Festuca_valesiaca_proba_matr

sum_psi_sites_fes <- apply(festuca_full.pred$psi.0.samples, c(1,3), sum)
psi_festuca <- colMeans(sum_psi_sites_fes)
sd_psi_fes <- apply(sum_psi_sites_fes, 2, sd)

total_presence <- vector("numeric",length=30)
for(i in 1:30){ #doesn't change according to the model type
  total_presence[i]<- sum(st_drop_geometry(festuca.eg_data[,i+34]))
  #print(sum(st_drop_geometry(taxa_all_info[,i+34])))
}

festuca_spi_yr_sum <- as.data.frame(colSums(festuca_spi, na.rm = T)[(ncol(festuca_spi)-28):ncol(festuca
colnames(festuca_spi_yr_sum) <- c("spi_sum")
festuca_spi_yr_sum <- rbind(festuca_spi_yr_sum, NA)

#mean zsamples across all mcmc samples
z_0_sum_all_times.full<- apply(festuca_full.pred$z.0.samples, c(1,3), sum) #summing to get #presence
```

```

mean_each_time.full <- apply(z_0_sum_all_times.full, 2, mean) #mean across all mcmc samples
sd_zsample_fes <- apply(z_0_sum_all_times.full, 2, sd)

sum_zsampels_reduced <- data.frame(Time =years_red, sum_zsamples = mean_each_time.full, presence=total)

ggplot(sum_zsampels_reduced[2:30,], aes(x = Time)) +
  geom_line(aes(y = sum_zsamples, color = "Pred Detections (Zsample)")) + # No space after "sum"
  geom_point(aes(y = sum_zsamples, color = "Pred Detections (Zsample)", shape = "Pred Detections (Zsample)")) +

  geom_line(aes(y = presence, color = "Presence")) +
  geom_point(aes(y = presence, color = "Presence", shape = "Presence")) +

  geom_line(aes(y = psi, color = "Occpuapncy Probability (Psi)")) +
  geom_point(aes(y = psi, color = "Occpuapncy Probability (Psi)", shape = "Occpuapncy Probability (Psi)")) +

  geom_line(aes(y=spi_sum, color="SPI results Festuca"))+
  geom_point(aes(y=spi_sum, color="SPI results Festuca", shape="SPI results Festuca"))+

  geom_errorbar(aes(y = sum_zsamples, ymin = sum_zsamples - sd_zsample, ymax = sum_zsamples + sd_zsample,
                    color = "blue", width = 0.2) +

  geom_errorbar(aes(y = psi, ymin = psi - sd_psi, ymax = psi + sd_psi),
                color = "green", width = 0.2) +

  labs(x = "Time", y = "Values", title = paste0("Line Graph of Zsamples ('Full' Model) & Presence over Time")) +
  scale_color_manual(values = c("Pred Detections (Zsample)" = "blue", "Presence" = "red", "Occpuapncy Probability (Psi)" = "green"),
                    name = "Legend") +
  scale_shape_manual(values = c("Pred Detections (Zsample)" = 16, "Presence" = 17, "Occpuapncy Probability (Psi)" = 18),
                    name = "Legend") +
  theme_minimal()

```

```

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

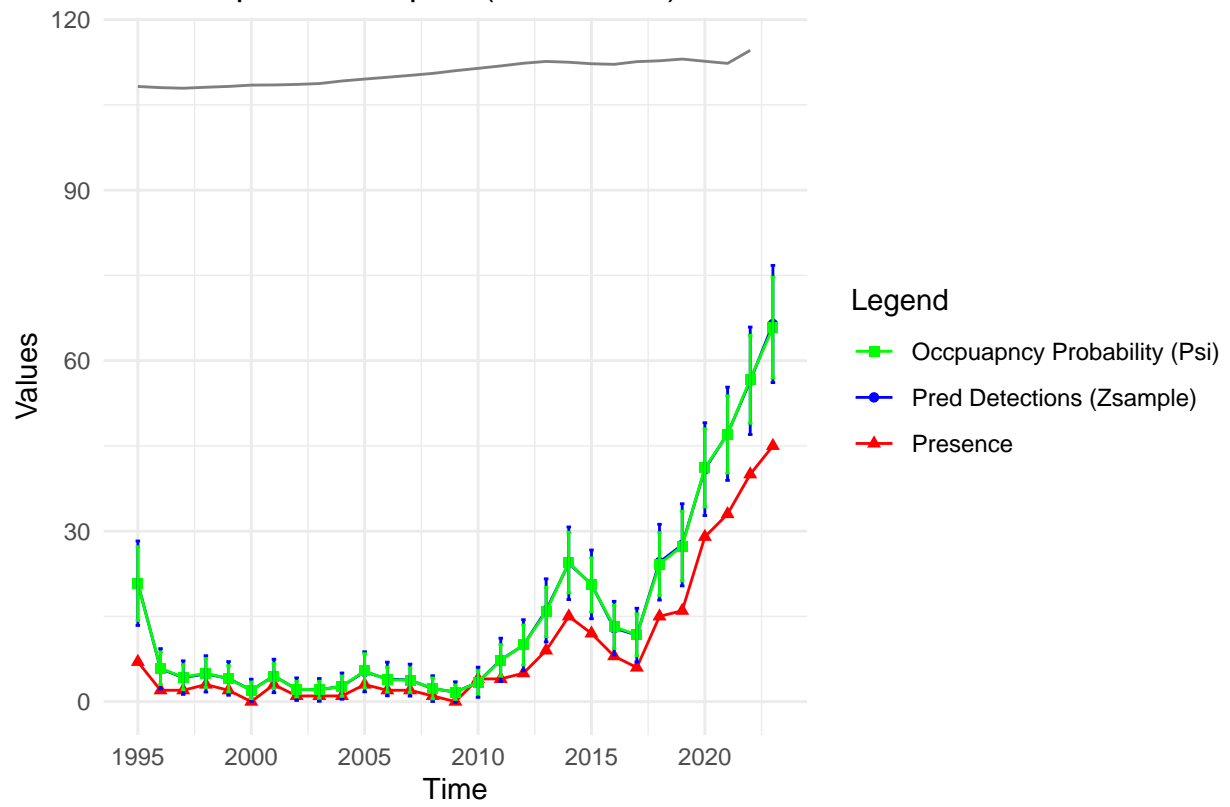
```

```

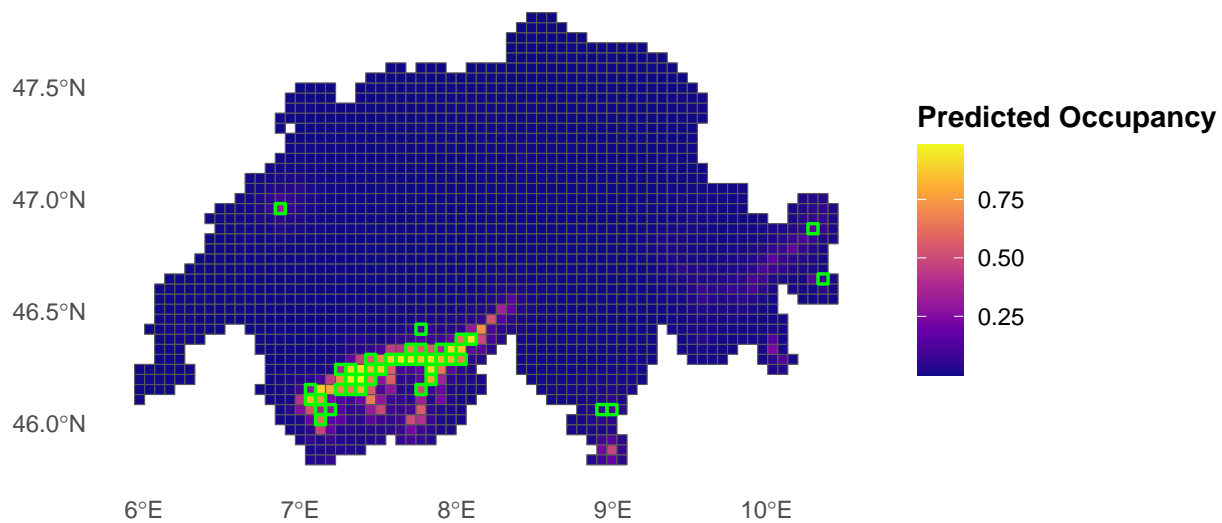
## Warning: Removed 29 rows containing missing values or values outside the scale range
## (`geom_point()`).

```

Line Graph of Zsamples ('Full' Model) & Presence over Time Festuca Val



## Festuca Valesiaca G. Response Variable with Model Predicted Oc



```
festuca_spi <- read.csv("C:/Users/IF_240606/Desktop/Projet/SPIcomp/festuca_spi_wcoords.csv")
festuca_spi <- st_as_sf(festuca_spi, coords = c("X","Y"), crs=21781)

ggplot(data = festuca.eg_data) +
  # First layer: Model predicted occupancy
  geom_sf(aes(fill = psi.pred_23_graph), alpha = 1, size = 0.3) +
  scale_fill_viridis_c(option = "plasma", name = "Predicted Occupancy") + # Viridis color for continuous

  # Second layer: Presence data with transparent fill and visible border
  geom_sf(data = fes_presences_nonzero, color = "green", fill = NA, size = 0.5, alpha = 0.7, linewidth=0.5) +

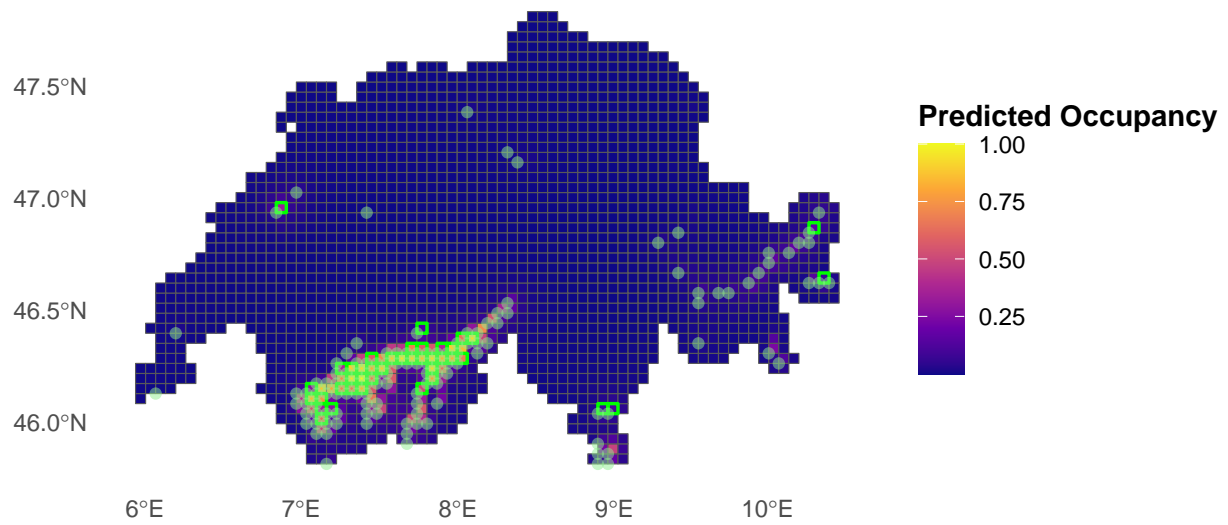
  geom_sf(data=festuca_spi, aes(fill =X2022), alpha=0.5, color="lightgreen")+

  #geom_sf(data=ws, color="pink", alpha=0.1, size=0.1)+

  # Customize theme and labels
  theme_minimal() +
  theme(
    panel.grid = element_blank(), # Remove grid for clarity
    legend.position = "right", # Place legend to the right for easy access
    legend.title = element_text(face = "bold"),
    plot.title = element_text(size = 14, face = "bold")
  ) +
  labs(
    title = "Graphical comparison SPI against BHM",
    fill = "Occupancy Prediction" # Label for the fill legend
  )
```

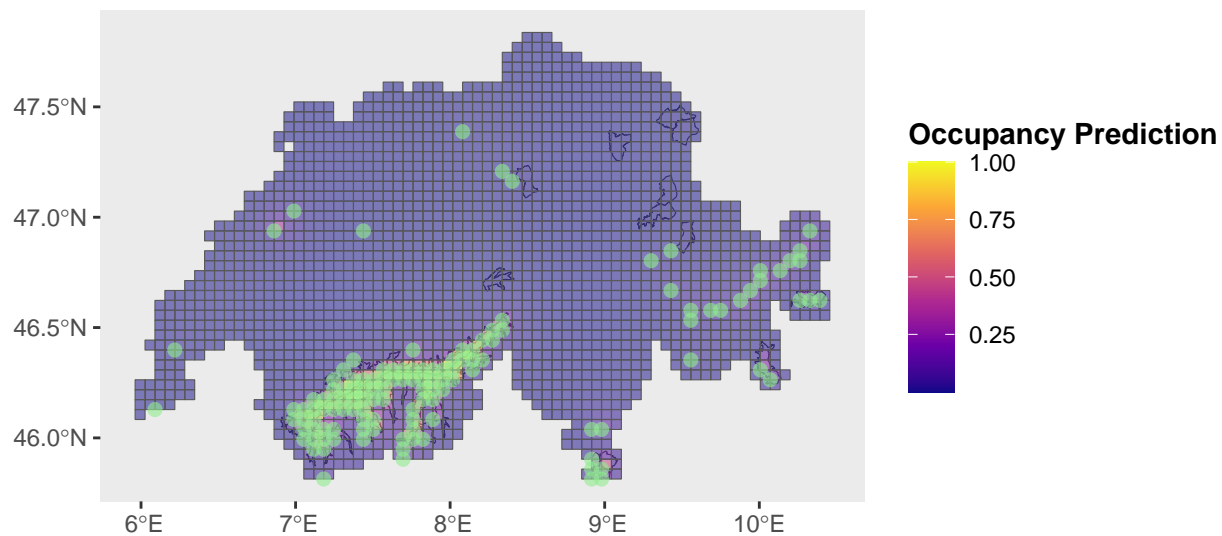
)

## Graphical comparison SPI against BHM



Donc nous avons plusieurs site où le SPI nous rend une prédiction quand nous n'avions pas de prédictions d'occupancy prédié par le MBH.

## Graphical comparison SPI against BHM



Le graphie indique qu'il y a certaines zones où nous avons de l'information dans l'atlas welten sutter mais qui nous pas de prédictions d'occupancy selon la méthode Swiss Plante Index.



## Saxifraga Hirculus:

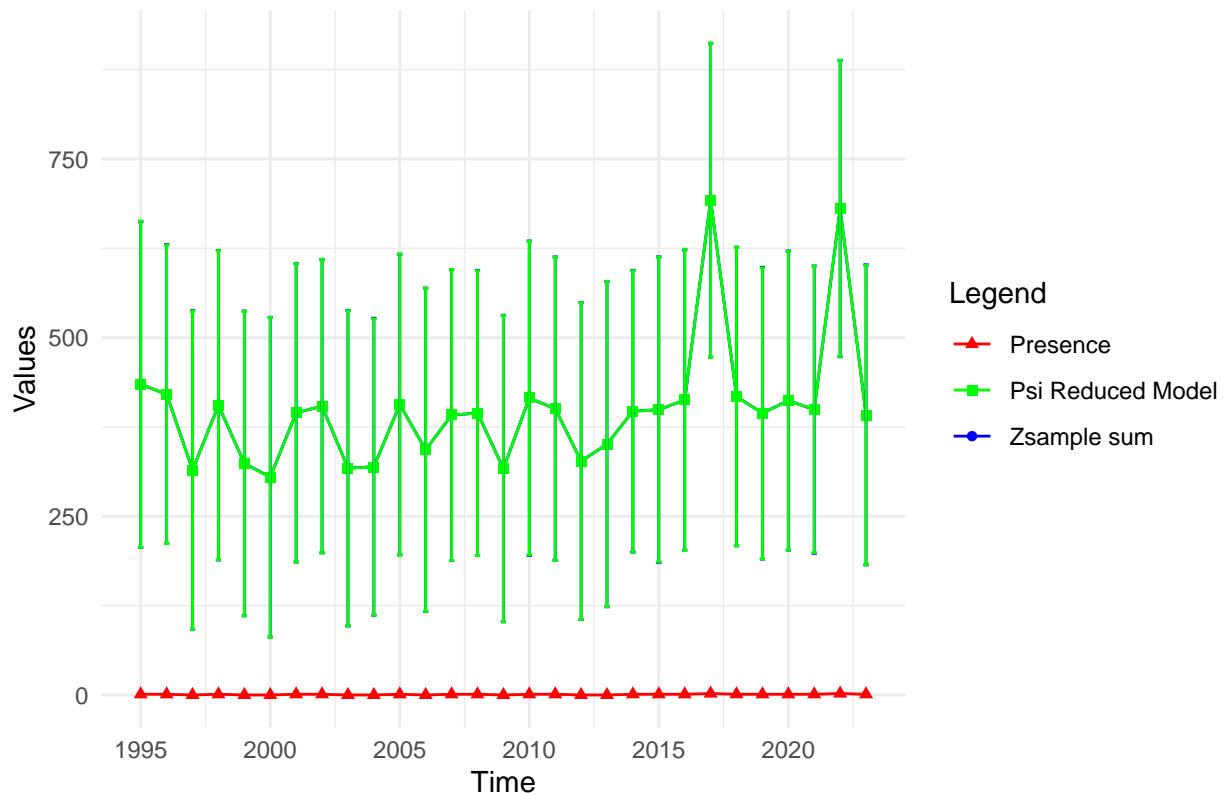
```
## List of 4
## $ y      : num [1:26, 1:30, 1] 1 1 1 1 1 1 1 1 1 1 ...
## $ occ.covs:List of 4
##   ..$ id_grid   : int [1:26] 647 913 914 983 1054 1063 1065 1125 1136 1206 ...
##   ..$ wsprop    : num [1:26] 0 0.252 0.206 0.227 0.276 ...
##   ..$ suitability: num [1:26] 13 0 31.8 0 0 ...
##   ..$ years     : num [1:26, 1:30] 1994 1994 1994 1994 1994 ...
## $ det.covs:List of 3
##   ..$ effort   : num [1:26, 1:30] 8.52 8.57 9.03 8.44 8.85 ...
##   ..$ id_grid  : int [1:26] 647 913 914 983 1054 1063 1065 1125 1136 1206 ...
##   ..$ years    : num [1:26, 1:30] 1994 1994 1994 1994 1994 ...
## $ coords   : num [1:26, 1:2] 2567500 2497500 2502500 2497500 2502500 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "X" "Y"

##
## Call:
## stPGOcc(occ.formula = occ_formula, det.formula = det_formula,
##   data = data_list_reduced, inits = inits_reduced, priors = priors,
##   n.batch = n.batch, batch.length = batch.length, n.omp.threads = 11,
##   ar1 = ar1, n.report = 50, n.burn = n.burn, n.thin = n.thin,
##   n.chains = n.chains)
##
## Samples per Chain: 5000
## Burn-in: 2000
## Thinning Rate: 12
## Number of Chains: 3
## Total Posterior Samples: 750
## Run Time (min): 0.2648
##
## Occurrence (logit scale):
##           Mean      SD    2.5%    50%  97.5%   Rhat ESS
## (Intercept) -0.0153 0.1226 -0.2472 -0.0159 0.2274 1.0065 750
## scale(suitability) 0.0480 0.1220 -0.1948 0.0463 0.2910 1.0028 750
## I(scale(suitability)^2) 0.0293 0.1222 -0.1999 0.0232 0.2743 1.0187 652
## scale(wsprop) 0.0025 0.1237 -0.2490 0.0060 0.2380 1.0385 750
## I(scale(wsprop)^2) -0.0337 0.1193 -0.2623 -0.0344 0.2009 1.0009 750
##
## Detection (logit scale):
##           Mean      SD    2.5%    50%  97.5%   Rhat ESS
## (Intercept) 0.1915 0.1149 -0.0382 0.1930 0.4248 1.0008 872
## scale(effort) 0.2529 0.1087 0.0442 0.2505 0.4618 1.0278 750
##
## Spatio-temporal Covariance:
##           Mean      SD    2.5%    50%  97.5%   Rhat ESS
## sigma.sq 36.1976 20.1601 11.4906 31.8819 87.0912 1.1448 63
## phi      0.0000 0.0000 0.0000 0.0000 0.0001 1.1082 185
## sigma.sq.t 13.3385 7.0791 4.4523 11.8080 31.2040 1.0590 71
## rho      0.0583 0.1771 -0.2640 0.0575 0.4065 1.0376 200

## List of 6
```

```
## $ z.0.samples : num [1:750, 1:1849, 1:30] 1 1 1 1 1 1 1 1 1 1 ...
## $ w.0.samples : 'mcmc' num [1:750, 1:1849] 4.99 4.06 -4.19 -6.86 -11.29 ...
## ..- attr(*, "mcpair")= num [1:3] 1 750 1
## $ psi.0.samples: num [1:750, 1:1849, 1:30] 1 1 0.998 0.992 0.56 ...
## $ run.time      : 'proc_time' Named num [1:5] 14.53 0.36 15.57 NA NA
## ..- attr(*, "names")= chr [1:5] "user.self" "sys.self" "elapsed" "user.child" ...
## $ call          : language predict.stPGOcc(object = out_reduced, X.0 = X.0, coords.0 = coords.full,
## $ object.class  : chr "stPGOcc"
## - attr(*, "class")= chr "predict.stPGOcc"
```

Line Graph of Zsamples (Reduced Model) & Presence over Time Saxifraga



Modèle “full” pour Saxifraga Hirculus L.:

```
model.eg.saxifraga_full <- readRDS(paste0("C:/Users/IF_240606/Desktop/Projet/",
"DYNAMICModels30/stPGOcc.Saxifraga.Model.Rdata"))
summary(model.eg.saxifraga_full)
```

```
##
## Call:
## stPGOcc(occ.formula = occ_formula, det.formula = det_formula,
## data = data_list, inits = inits, priors = priors, n.batch = n.batch,
## batch.length = batch.length, n.omp.threads = 11, ar1 = ar1,
## n.report = 50, n.burn = n.burn, n.thin = n.thin, n.chains = n.chains)
##
## Samples per Chain: 5000
## Burn-in: 2000
```

```
## Thinning Rate: 12
## Number of Chains: 3
## Total Posterior Samples: 750
## Run Time (min): 13.258
##
## Occurrence (logit scale):
##           Mean      SD    2.5%    50%  97.5%   Rhat ESS
## (Intercept)   -0.0593 0.1206 -0.2970 -0.0577 0.1680 1.0334 159
## scale(suitability) 0.0223 0.1172 -0.2106 0.0212 0.2444 1.0212 160
## I(scale(suitability)^2) 0.1703 0.0619 0.0435 0.1734 0.2863 1.0063 150
## scale(wsprop) 0.1362 0.1241 -0.1090 0.1346 0.3593 1.0099 667
## I(scale(wsprop)^2) 0.1389 0.0323 0.0761 0.1383 0.2055 1.0907 108
##
## Detection (logit scale):
##           Mean      SD    2.5%    50%  97.5%   Rhat ESS
## (Intercept) 0.0614 0.1154 -0.1691 0.0663 0.2811 1.0035 1143
## scale(effort) 0.2101 0.1074 -0.0060 0.2100 0.4212 1.0052 750
##
## Spatio-temporal Covariance:
##           Mean      SD    2.5%    50%  97.5%   Rhat ESS
## sigma.sq 3.6819 1.0472 1.9814 3.5598 5.9392 1.3166 19
## phi      0.0004 0.0001 0.0002 0.0005 0.0006 1.1215 35
## sigma.sq.t 33.4718 11.0927 17.0202 31.6904 57.6147 1.0325 73
## rho      0.8192 0.0502 0.6943 0.8260 0.8970 1.0138 144
```

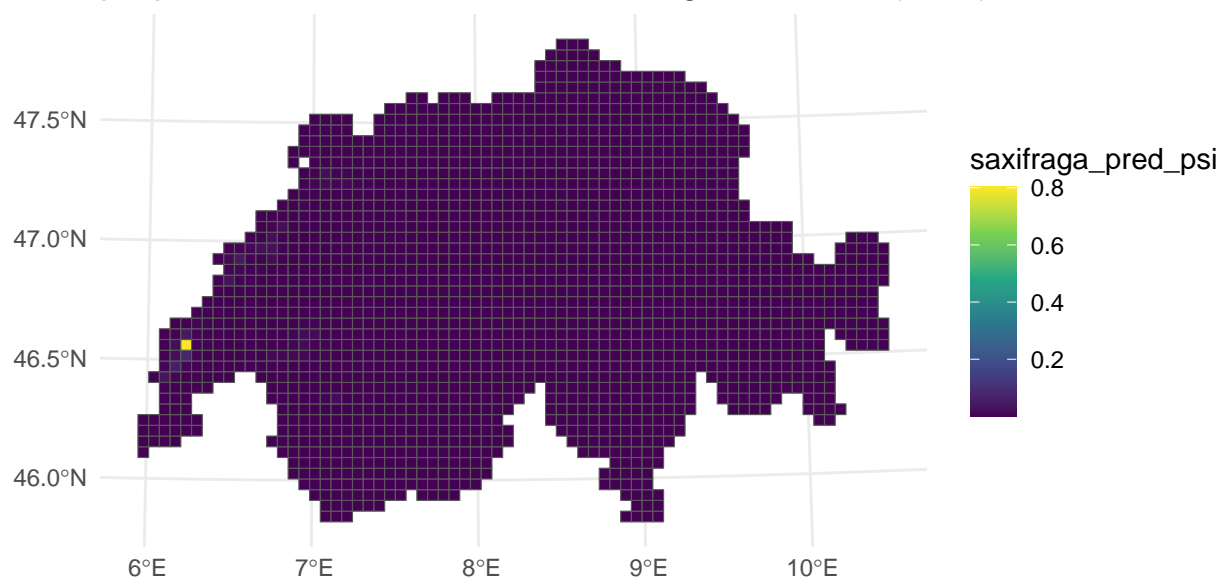
```
saxifraga_full.pred <- readRDS(paste0("C:/Users/IF_240606/Desktop/Projet/",
  "DynamicModels30/stPGOcc_", "Saxifraga2", "_outpred.Rdata"))
str(saxifraga_full.pred)
```

```
## List of 6
## $ z.0.samples : num [1:750, 1:1849, 1:30] 0 0 0 0 0 0 0 0 0 0 ...
## $ w.0.samples : 'mcmc' num [1:750, 1:1849] -4.11 -5.31 -3.53 -5.63 -4.59 ...
## ..- attr(*, "mcpair")= num [1:3] 1 750 1
## $ psi.0.samples: num [1:750, 1:1849, 1:30] 2.81e-04 7.50e-05 6.09e-04 7.53e-05 2.19e-04 ...
## $ run.time : 'proc_time' Named num [1:5] 5.99 0.24 6.95 NA NA
## ..- attr(*, "names")= chr [1:5] "user.self" "sys.self" "elapsed" "user.child" ...
## $ call : language predict.stPGOcc(object = out, X.0 = X.0, coords.0 = coords.full, t.cols =
## $ object.class : chr "stPGOcc"
## - attr(*, "class")= chr "predict.stPGOcc"
```

```
saxifraga.eg_data$saxifraga_pred_psi <- apply(saxifraga_full.pred$psi.0.samples[,
  , 30], 2, mean)
```

```
ggplot(saxifraga.eg_data) + geom_sf(aes(fill = saxifraga_pred_psi)) +
  scale_fill_viridis_c() + labs(title = "psi prediction for 'full' model of Saxifraga Hirculus l. (2017)")
theme_minimal()
```

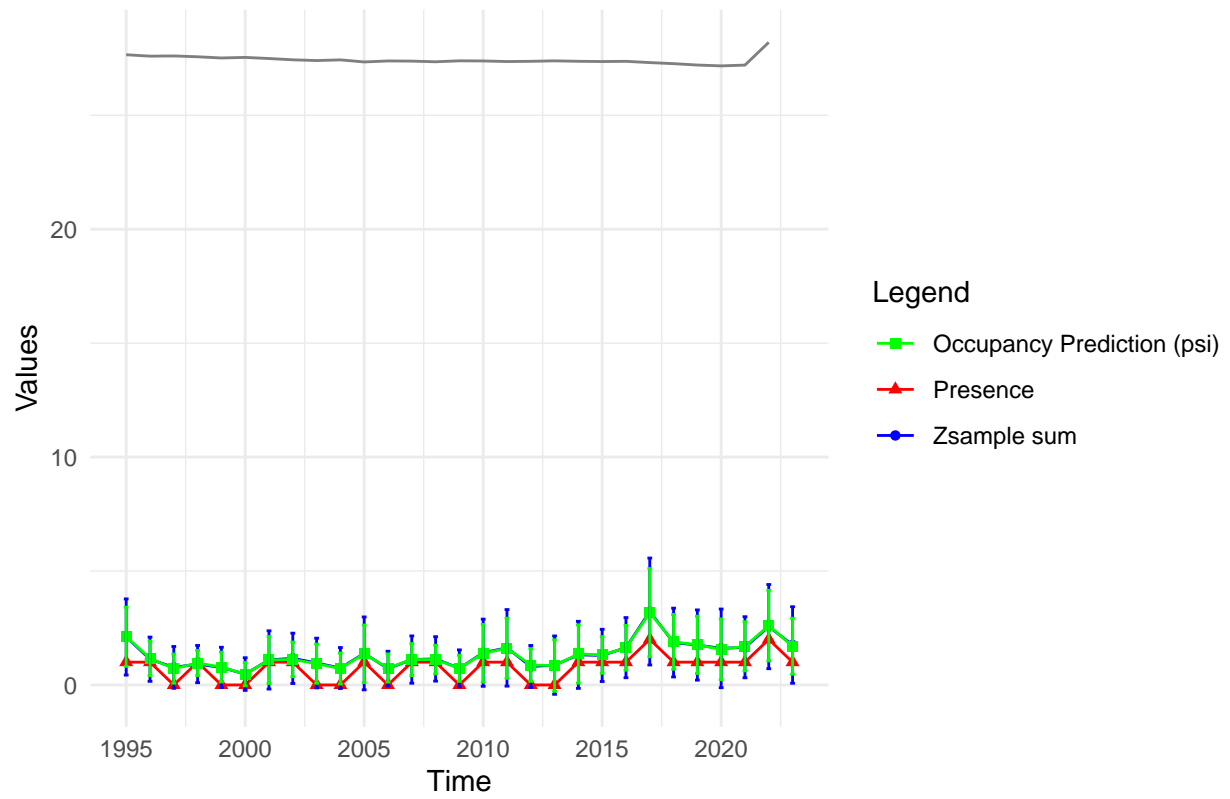
psi prediction for 'full' model of Saxifraga Hirculus I. (2023)



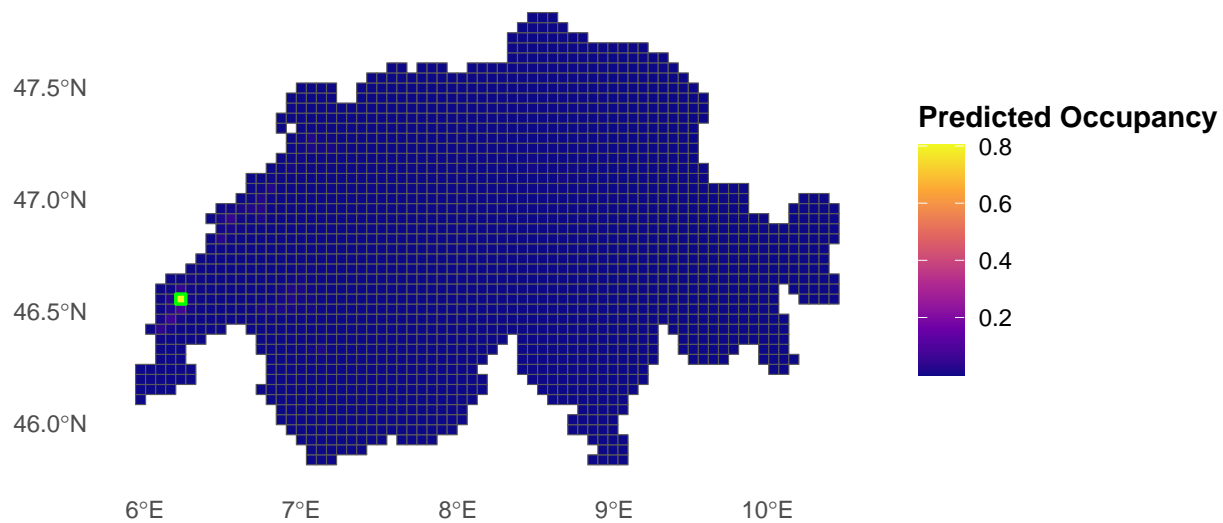
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 29 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Line Graph of Zsamples & Presence ('full' model) over Time Saxifraga Hircu



## **Festuca Valesiaca G. Response Variable with Model Predicted Oc**



## Nombre de carrés par modèle et difference entre présences et présences prédites

Espèce	Squares.model.full	Z.sample.diff
Medicago Lupulina l.	1603	4435
Festuca Valesiaca g.	1744	192
Saxifraga Hirculus l.	1839	32

## Nombre de carrés par modèle 'réduit' et difference entre présences et présences prédites

Espèce	Squares.model.reduced	Z.sample.diff.reduced
Medicago Lupulina l.	1276	4098
Festuca Valesiaca g.	105	8797
Saxifraga Hirculus l.	26	13262

**Discussion** Si nous comparons les résultats des prédictions de présence des deux types de modèles, pour chaque espèce, nous voyons très clairement que, lorsqu'on modélise une espèce rare ou même localement rare, le modèle a beaucoup de mal à prédire un nombre de présences raisonnable (dans le cas du modèle réduit).

La qualité des estimateurs des modèles: En général, nos échantillons MCMC pour chaque estimateur ont tendance à converger, et la taille de chaque échantillon (ESS) est considérable. Cela dit, dans le cas de Saxifraga (modèle 'full'), nos estimateurs spatio-temporels ne sont pas fiables (Rhat très élevé), ce qui est également évident en observant les ESS respectifs. Inversement, les estimateurs spatio-temporels pour Festuca valesiaca sont plus fiables dans le cas du modèle réduit.

Nos résultats, en général, sont trop corrélés avec les variables réponses, ce qui n'est pas bon signe, car cela démontre l'incapacité du modèle à corriger les présences en fonction de nos covariables. De plus, le modèle prédit constamment une population plus importante pour chaque espèce que celle observée dans la base de données. Cela n'est pas déraisonnable, même si, souvent, les quantités de prédiction paraissent trop élevées.

Donc, si nous voulions aller plus loin et faire du multi-espèces avec deux ou trois espèces ayant généralement le même niveau de rareté et une intersection géographique en Suisse, cela pourrait nous donner une autre perspective sur la valeur du modèle. Cependant, comme ce modèle a été conçu dans l'intention de pouvoir être évalué contre le Swiss Plant Index, qui traite les espèces individuellement, il faudrait rester conscient de cette distinction.

Discussion sur certaines autres formulations du modèle qui ont été testées:

-Modèle sans suitability.

-Modèle à échelle 2kmx2km.

-Modèle qui n'a pas été entraîné avec les données avant 1994.

-Modèle sans termes de standardisation dans formule d'occupancy et de detection.

Toutes les approches ont été testées sur Festuca Valesiaca et n'ont pas été capable de corriger la corrélation de nos tendances avec la variable réponse.

### Points à améliorer

1. Refaire les modèles "réduits" en étant moins strict sur la sélection de carré, en particulier pour les espèces rares.
2. Etudier et corriger l'autocorrélation spatiale

3. Être plus précis dans le choix de la distribution des priors.
4. Affiner encore plus les périodes de temps/les dimensions de chaque site (2x2km, 1x1km)
5. Rajouter plus de covariables
6. Plus grand nombre de MCMC samples par modèle (plus de chaînes, longueur plus grande de chaque chaîne)

### **Biobibliographie/Outils pratiques supplémentaires**

<https://doserlab.com/files/spoccupancy-web/articles/spacetimemodelshtml#fitting-multi-season-spatial-occupancy-models-with-stpgocc>

[https://pmarchand1.github.io/ECL8202/notes\\_cours/09-Modeles\\_hierarchiques\\_bayesiens.html](https://pmarchand1.github.io/ECL8202/notes_cours/09-Modeles_hierarchiques_bayesiens.html)

<https://doserlab.com/files/spoccupancy-web/articles/spacetimemodelshtml#modelDescription>

<https://doserlab.com/files/spoccupancy-web/articles/modelconsiderations#specifying-the-prior-for-the-spatial-decay-parameter-phi-in-spatial-occupancy-models>

<https://www.stat.berkeley.edu/~paciorek/research/techVignettes/techVignette5.pdf>