

# TD Fouille de données spatio-temporelles - Modélisation des dynamiques spatio-temporelles des abondances des vecteurs du paludisme en Côte d'Ivoire avec le langage de programmation R

Paul Taconet, IRD

Janvier 2023 - Révison Juin 2024

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Objectifs pédagogiques de l'exercice . . . . .	2
1.2	Présentation du contexte et des données entomologiques . . . . .	3
<b>2</b>	<b>Importer et préparer les données pour la modélisation</b>	<b>5</b>
2.1	Importer et préparer les données entomologiques . . . . .	5
2.2	Importer et préparer les données météorologiques . . . . .	6
2.3	Importer et préparer les données paysagères . . . . .	13
<b>3</b>	<b>Modéliser</b>	<b>17</b>
3.1	Visualiser les données et leurs associations . . . . .	18
3.2	Modélisation explicative . . . . .	20
3.3	Modélisation exploratoire . . . . .	21
3.4	Modélisation prédictive . . . . .	24
<b>4</b>	<b>Pour aller plus loin</b>	<b>26</b>

# 1 Introduction

## 1.1 Objectifs pédagogiques de l'exercice

Ce document est un exercice à destination de personnes souhaitant approfondir leurs connaissances en manipulation de données spatiales et spatio-temporelles sur R, ainsi qu'en modélisation statistique. Nous abordons des notions avancées de SIG (manipulation de données d'occupation du sol, extraction de séries spatio-temporelles au format NetCDF, modélisation spatiale) à travers un cas d'étude lié à la santé publique : la modélisation des dynamiques spatio-temporelles des abondances des vecteurs du paludisme en fonction des conditions environnementales. Notre zone d'étude est la région de Korhogo, située au nord de la Côte d'Ivoire.

Cet exercice requiert le logiciel R. Après une rapide présentation du cas d'étude théorique, nous présentons les données utilisées dans l'exercice, puis nous déroulons l'exercice. Afin de contrôler le bon avancement du tutoriel, des questions sont régulièrement posées tout au long de l'exercice. Les réponses sont disponibles en fin de document.

**Prérequis pour aborder sereinement le document :** Connaissances en SIG, connaissances de base dans le langage de programmation R

**Notions et concepts techniques abordés :** extraction de données spatio-temporelles, séries spatio-temporelles issues d'images d'observation de la Terre, NetCDF, modélisation statistique

**Nota bene :** le code R présenté dans ce document n'est pas toujours nécessairement optimisé en terme de performances. A l'image du document dans son ensemble, il a une vocation avant tout pédagogique.

## 1.2 Présentation du contexte et des données entomologiques

### 1.2.1 Contexte

L'objectif principal de l'exercice est de modéliser l'abondance spatio-temporelle de vecteurs du paludisme en fonction des conditions environnementales paysagères et météorologiques - autrement dit, d'identifier les déterminants environnementaux de la présence et de l'abondance des vecteurs et de prédire les abondances en fonction de ces mêmes conditions environnementales.

Les conditions météorologiques (températures, précipitations) et paysagères (utilisation, occupation du sol), impactent de nombreux traits de vie des vecteurs : émergence, croissance, survie, dispersion, activité, etc. Par exemple, la température affecte le moustique, à chaque étape de son cycle de vie (croissance larvaire, survie des adultes, etc.). De leur côté, les précipitations remplissent ou créent les gîtes larvaires et expliquent ainsi, en partie, la saisonnalité de l'abondance de certaines espèces d'anophèles. La fréquence des précipitation, leur abondance, leur durée, sont donc des paramètres essentiels pour expliquer les densités des vecteurs.

Comprendre de quelle manière l'environnement impacte la distribution et la densité des anophèles, et être en mesure de prédire ces densités dans l'espace et dans le temps, peut *in fine* aider à concevoir et déployer des interventions de lutte anti-vectorielle.

### 1.2.2 Source des données

L'exercice proposé utilise des données de terrain collectées dans le cadre du projet *REACT : Gestion de la résistance aux insecticides au Burkina Faso et en Côte d'Ivoire : recherche sur les stratégies de lutte anti-vectorielle*, mené en partenariat entre l'Institut de Recherche pour le Développement (IRD, France), l'Institut de Recherche en Sciences de la Santé (IRSS, Burkina Faso) et l'Institut Pierre Richet (IPR, Côte d'Ivoire). Ce projet était financé par L'Initiative 5% (Expertise France). L'objectif principal de ce projet, dont la phase de terrain s'est déroulée entre les années 2016 et 2018, était d'évaluer l'impact de l'utilisation de mesures de lutte anti-vectorielles complémentaires à la moustiquaire imprégnée d'insecticide, sur la transmission et l'épidémiologie du paludisme à travers un essai randomisé contrôlé (ERC). A cette fin, deux zones d'études ont été sélectionnées dans deux pays d'Afrique de l'ouest : le Burkina Faso (BF) et la Côte d'Ivoire (CI). Ces deux pays sont situés en zone endémiques du paludisme à *P. falciparum*.

Chaque zone d'étude du projet REACT couvre environ la surface d'un district sanitaire rural ouest-africain (~2500 km<sup>2</sup>). Il s'agit de zones principalement rurales. Pour le projet REACT, un total de 55 villages (27 au Burkina Faso, 28 en Côte d'Ivoire) a été sélectionné au sein de ces zones pour mener l'ERC selon les critères suivants : accessibilité pendant la saison des pluies, 200 à 500 habitants par village, et distance entre les villages supérieure à 2 km. La figure 1 présente la localisation géographique des zones et des villages sélectionnés ; ainsi que le chronogramme de collectes de données effectuées dans le cadre du projet REACT.

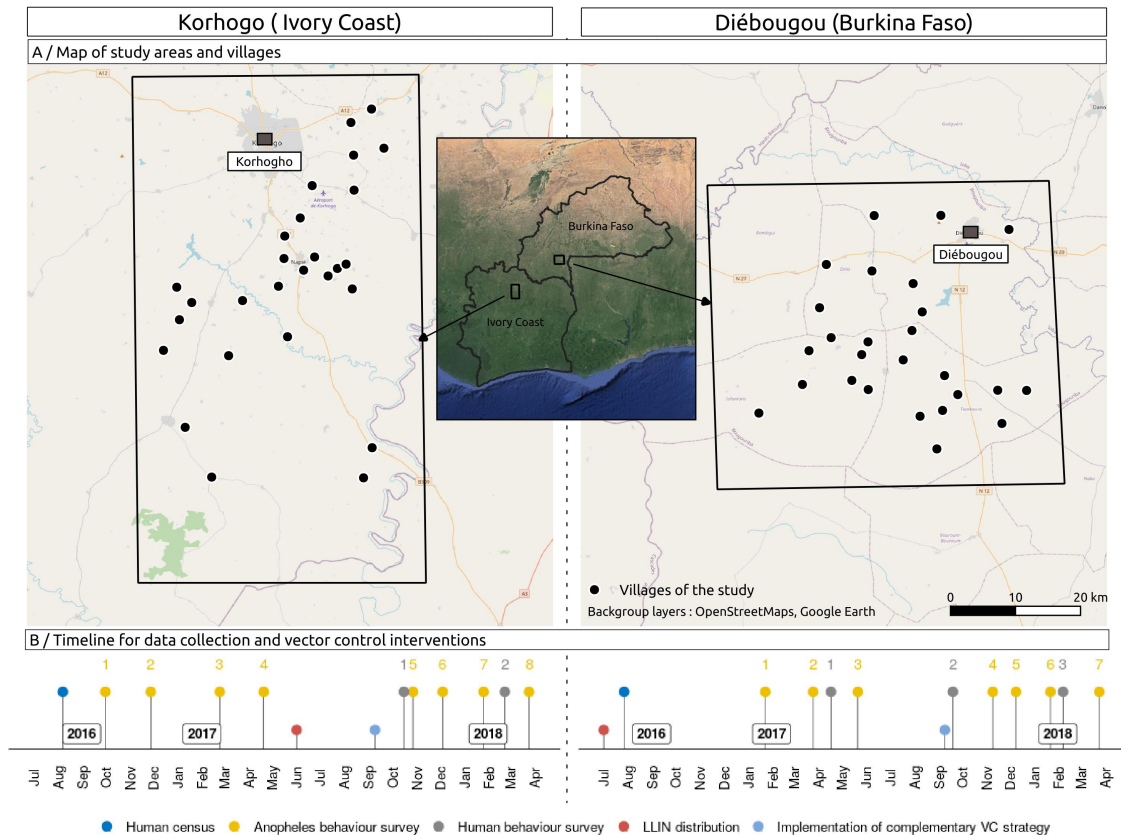


Figure 1: Projet REACT : zones d'étude, villages et dates de collectes des données

Dans cet exercice, nous allons nous focaliser sur la zone Ivoirienne du projet REACT. Cette zone couvre la région de Korhogo, au nord du pays, en région bioclimatique soudanienne. Le climat y est caractérisé par une saison sèche d'octobre à avril (incluant une période 'froide' de décembre à février et une période 'chaude' de mars à avril) et une saison pluvieuse de mai à septembre. La végétation naturelle est dominée par la savane arborée parsemée de forêts ripicoles.

Dans le cadre du projet REACT, plusieurs enquêtes entomologiques ont été effectuées dans chaque village au cours des 2 années du projet. Les moustiques ont été collectés en utilisant la technique de la capture sur sujet humain, de 17h00 à 09h00. Les anophèles ont ensuite été identifiés à l'espèce. Sur la zone ivoirienne, deux espèces/genres d'anophèles principales ont été identifiées : *An. gambiae s.l.* et *An. funestus*. Dans cet exercice, afin de faciliter les traitements, nous allons conserver et nous focaliser sur un seul genre : *An. gambiae s.l.*

L'exercice consiste, en partant d'un simple tableau contenant le nombre de moustiques collectés ainsi que les coordonnées géographiques et les dates de collecte, à :

- Identifier et collecter des données météorologiques et paysagères dans la région et aux dates de collecte, puis constituer des variables statistiques pertinentes à partir de ces données,
- Générer des **modèles descriptifs** (ou exploratoires) de l'abondance des moustiques,
- Générer des **modèles explicatifs** de l'abondance des moustiques,
- Générer des **modèles prédictifs** de l'abondance des moustiques.

C'est parti !

## 2 Importer et préparer les données pour la modélisation

### 2.1 Importer et préparer les données entomologiques

Le tableau contenant le nombre de moustiques collectés ainsi que les coordonnées géographiques et les dates de de collecte est disponible sous `entomological_data.csv`. Ouvrons le sous R :

```
entomological_data <- read.csv("data/CI/entomological_data.csv")

head(entomological_data)
```

```
## mission      date village      X      Y      n
## 1         1 2016-09-21    GUE -5.548970 9.271823 120
## 2         1 2016-09-21    LOK -5.605390 9.186830 156
## 3         1 2016-09-21    PEN -5.514849 9.253527  84
## 4         1 2016-09-22    KOL -5.524137 9.288387 131
## 5         1 2016-09-23    KAG -5.535534 9.282156 165
## 6         1 2016-09-23    NOK -5.614132 9.251798 188
```

🔧 Décrivez la table en une ou deux phrases simples (granulométrie des données, type d'informations qui y sont disponibles, etc.) .

🔧 Créez un histogramme des abondances d'anophèles capturés (indice : utilisez la fonction `hist()` sur la bonne colonne de la table `entomological_data`).

La première étape du travail consiste à convertir cet objet (`data.frame`) au format `sf`. `sf` est la librairie R de référence pour manipuler des données géographiques vectorielles (points, lignes, polygones) sur R.

```
library(sf)
library(dplyr)

entomological_data <- entomological_data %>%
  st_as_sf(coords = c("X", "Y"), crs = 4326) %>% # conversion au format sf
  mutate(date = as.Date(date)) # transformation de la colonne Date au type Date

entomological_data
```

```
## Simple feature collection with 231 features and 4 fields
## Geometry type: POINT
## Dimension:      XY
## Bounding box:   xmin: -5.779179 ymin: 8.884571 xmax: -5.471034 ymax: 9.505276
## Geodetic CRS:   WGS 84
## First 10 features:
## mission      date village      n      geometry
## 1         1 2016-09-21    GUE 120 POINT (-5.54897 9.271823)
## 2         1 2016-09-21    LOK 156 POINT (-5.60539 9.18683)
## 3         1 2016-09-21    PEN  84 POINT (-5.514849 9.253527)
## 4         1 2016-09-22    KOL 131 POINT (-5.524137 9.288387)
## 5         1 2016-09-23    KAG 165 POINT (-5.535534 9.282156)
```

```
## 6      1 2016-09-23      NOK 188 POINT (-5.614132 9.251798)
## 7      1 2016-09-23      NOT 105 POINT (-5.617933 9.257615)
## 8      1 2016-09-24      LAG 116 POINT (-5.567505 9.298098)
## 9      1 2016-09-26      YEN 196 POINT (-5.587993 9.352712)
## 10     1 2016-09-27      TAK 143 POINT (-5.60531 9.325821)
```

BONUS (facultatif) : Nous pouvons facilement cartographier les points de collecte de moustiques à l’aide la librairie `mapview` :

```
library(mapview)

mapview(entomological_data, legend = F)
```

## 2.2 Importer et préparer les données météorologiques

Dans des régions où les stations météorologiques sont rares (comme c’est le cas dans les milieux ruraux ouest-africains), les images satellitaires sont une source précieuse de données météo. Il existe de très nombreuses sources de données météorologiques satellitaires. Pour cet exercice, nous allons utiliser des données satellitaires produites par la NASA : des données de température de surface recueillies par l’instrument [MODIS](#) embarqué à bord du satellite Terra de la NASA, et des données de précipitations produites par la mission [Global Precipitation Measurement](#).

En particulier, nous allons utiliser les collections suivantes :

- la collection [MOD11A2.061](#) : Température de surface. Couverture : mondiale ; résolution spatiale : 1 km ; résolution temporelle : 8 jours
- la collection [GPM\\_3IMERGDF.07](#) : Précipitations. résolution spatiale : 0.1° (~10 km) ; résolution temporelle : 1 jour

### 2.2.1 Importer les données

Il existe de nombreux moyens d’accéder à ces données météorologiques satellitaires. Pour cet exercice, nous allons y accéder en utilisant la librairie R `modisfast`. `modisfast` est une librairie qui permet de télécharger les données MODIS et plusieurs autres sources de données d’observation de la Terre d’une manière efficace et rapide : en les échantillonnant lors de la phase de téléchargement (spatialement, temporellement et dimensionnellement) grâce au protocole [OPeNDAP](#).

`modisfast` requiert comme paramètres :

- une collection de données d’intérêt
- une ou plusieurs bandes d’intérêt pour cette collection
- une zone géographique d’intérêt de type `sf` POLYGON
- une période de temps d’intérêt (borne temporelle inférieure et supérieure)

Dans le cadre de cet exercice, nos collections et bandes d’intérêt sont les suivantes :

- collection “MOD11A2.061”, bandes “LST\_Day\_1km” et “LST\_Night\_1km”
- collection “GPM\_3IMERGDF.07”, bande “precipitation”

La zone géographique et les dates d'intérêts doivent couvrir un peu plus large que l'ensemble des points et des dates de collecte (vous comprendrez pourquoi par la suite). Nous allons donc générer une région d'intérêt couvrant les points de collecte, puis nous allons élargir un peu cette région (de 3 km dans toutes les directions). De même, nous allons générer des dates d'intérêt couvrant les dates de collecte, puis nous allons élargir un peu ces dates (de 30 jours avant la première collecte).

Pour la région d'intérêt :

```
# calculons les coordonnées (N, S, E, O) délimitant les points de collecte
roi <- sf::st_bbox(entomological_data)

# Etendons la région. Pour cela, utilisons la fonction expand_bbox() qui permet d'étendre
# la zone d'intérêt d'une distance donnée dans les directions N-S et E-W
# (merci à @Chrisjb pour cette fonction)
source("https://raw.githubusercontent.com/Chrisjb/basemapR/master/R/expand_bbox.R")
roi <- roi %>%
  expand_bbox(.,3000,3000) %>% # 3000 m dans toutes les directions
  sf::st_as_sfc() %>%
  sf::st_sf()

# Enfin, donnons un nom à la zone d'intérêt
roi$id = "korhogo"

roi
```

```
## Simple feature collection with 1 feature and 1 field
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: -5.8063 ymin: 8.857588 xmax: -5.443927 ymax: 9.53226
## Geodetic CRS: WGS 84
## geometry id
## 1 POLYGON ((-5.8063 8.857588,... korhogo
```

```
mapview(list(roi, entomological_data), legend = F)
```

et pour les dates d'intérêt :

```
# Créons un vecteur de 2 dates, contenant la date minimale
# (30 jours avant la 1ere collecte) et la date maximale
# (date de la dernière collecte)
time_range <- c(min(entomological_data$date) - 30, max(entomological_data$date))

time_range
```

```
## [1] "2016-08-22" "2018-04-03"
```

Nous avons à présent défini l'ensemble des paramètres nécessaires pour télécharger les données météorologiques avec la librairie `modisfast`. Procédons donc au téléchargement :

*Note 1: Pour pouvoir accéder à ces données, il faut créer un compte utilisateur Earthdata ici : <https://urs.earthdata.nasa.gov/>*

Note 2 : si vous rencontrez des problèmes pour télécharger ces données, elles sont disponibles dans le dossier 'data/meteorological\_data'.

```
library(modisfast)

# Insérer votre nom d'utilisateur et mot de passe Earthdata
# dans la fonction suivante :
log <- modisfast::mf_login(credentials = c(Sys.getenv("earthdata_un"),
      Sys.getenv("earthdata_pw")))

# Les fonctions suivantes génèrent les URLs pour
# télécharger les données :
urls_mod11a2 <- modisfast::mf_get_url(collection = "MOD11A2.061",
      variables = c("LST_Day_1km", "LST_Night_1km"), roi = roi,
      time_range = time_range)

urls_gpm <- modisfast::mf_get_url(collection = "GPM_3IMERGDF.07",
      variables = c("precipitation"), roi = roi, time_range = time_range)

# Les fonctions suivantes téléchargent les données en
# local, dans le dossier 'data/meteorological_data' :
res_dl_modis <- modisfast::mf_download_data(urls_mod11a2, path = "data/meteorological_data")
res_dl_gpm <- modisfast::mf_download_data(urls_gpm, path = "data/meteorological_data")
```

Les données météorologiques sont à présent téléchargées ! Importons les et visualisons les dans R. Pour cela, nous allons utiliser la librairie **terra**. Si **sf** fait référence pour la manipulation de données vectorisées, **terra** fait référence pour la manipulation des données rasterisées (mais il en existe de nombreuses autres : **stars**, **ncdf4**, etc.).

```
library(terra)

# Importons les données MODIS au format SpatRast à l'aide
# de la fonction 'mf_import_data' de la librairie modisfast
# :
modis_ts <- modisfast::mf_import_data(path = "data/meteorological_data/korhogo/MOD11A2.061",
      collection_source = "MODIS")

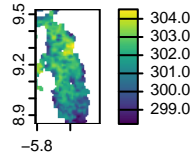
# De même pour les données GPM :
gpm_ts <- modisfast::mf_import_data(path = "data/meteorological_data/korhogo/GPM_3IMERGDF.07",
      collection_source = "GPM")
```

Visualisons les données ainsi que leur structure :

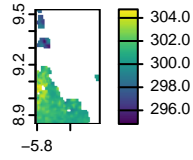
```
terra::plot(modis_ts)
```



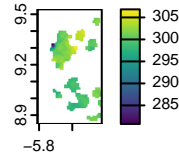
**2016-08-20**



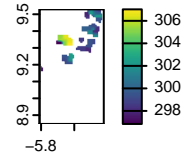
**2016-08-28**



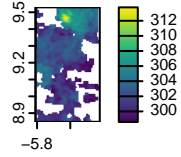
**2016-09-05**



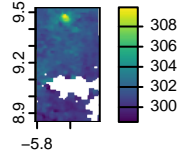
**2016-09-13**



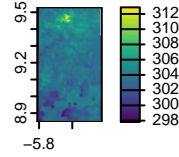
**2016-09-21**



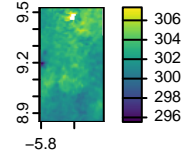
**2016-09-29**



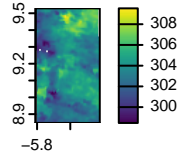
**2016-10-07**



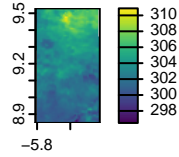
**2016-10-15**



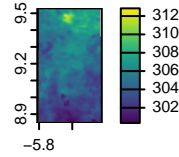
**2016-10-23**



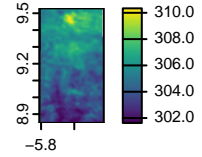
**2016-10-31**



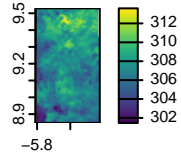
**2016-11-08**



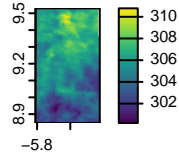
**2016-11-16**



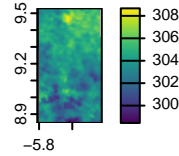
**2016-11-24**



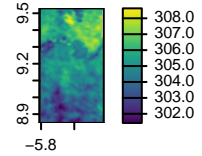
**2016-12-02**



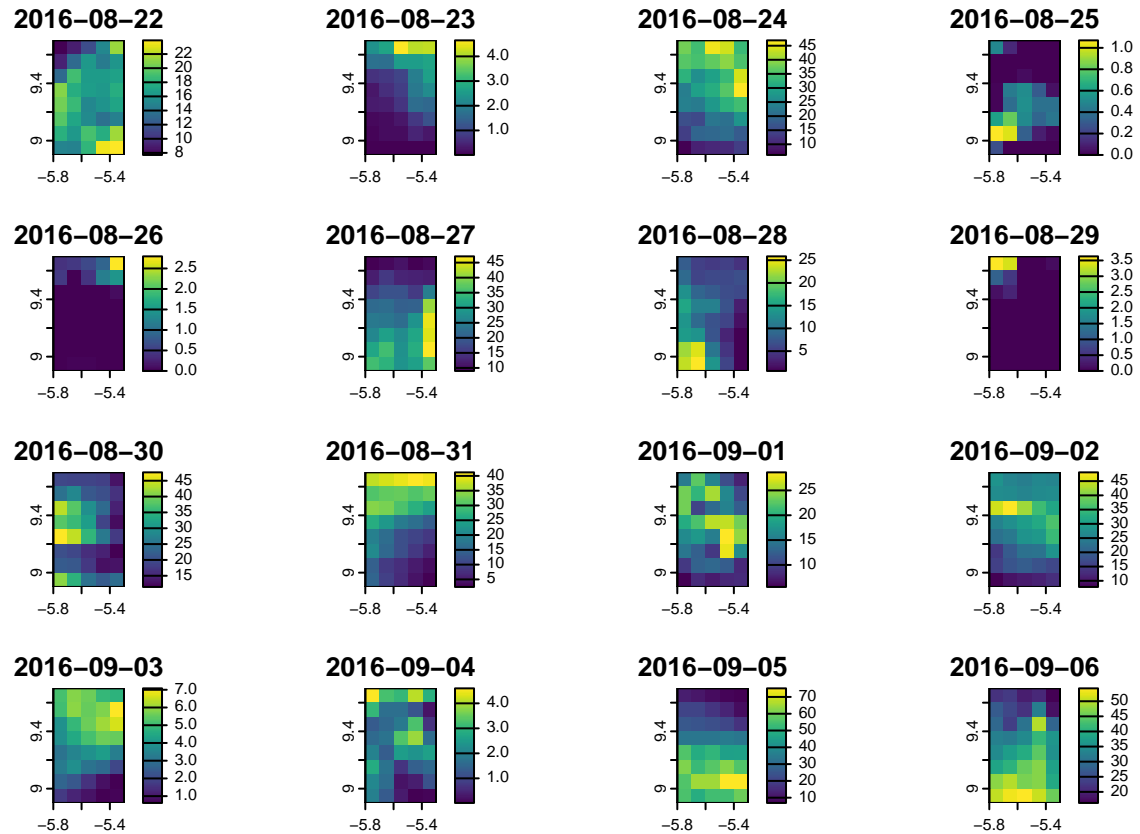
**2016-12-10**



**2016-12-18**



```
terra::plot(gpm_ts)
```



```
modis_ts
```

```
## class      : SpatRaster
## dimensions  : 81, 46, 150 (nrow, ncol, nlyr)
## resolution  : 0.008415423, 0.008415423 (x, y)
## extent     : -5.809327, -5.422217, 8.847517, 9.529167 (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source(s)   : memory
## names       : LST_D~1km_1, LST_D~1km_2, LST_D~1km_3, LST_D~1km_4, LST_D~1km_5, LST_D~1km_6, ...
## min values  :      298.22,      294.8208,      280.78,      296.84,      298.6000,      298.5273, ...
## max values  :      304.52,      304.7400,      306.84,      307.02,      313.7795,      309.9034, ...
## unit        :          K,          K,          K,          K,          K,          K, ...
## time (days) : 2016-08-20 to 2018-03-30
```


```
gpm_ts
```

```
## class      : SpatRaster
## dimensions  : 8, 5, 590 (nrow, ncol, nlyr)
## resolution  : 0.1, 0.1 (x, y)
## extent     : -5.8, -5.3, 8.9, 9.7 (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source(s)   : memory
## varname     : precipitation (Daily mean precipitation rate (combined microwave-IR) estimate. Formerly,
```

```
## names      : preci~ation, preci~ation, preci~ation, preci~ation, preci~ation, preci~ation, ...
## min values :      7.705,      0.005,      6.210,      0.00,      0.000,      8.99, ...
## max values :     23.940,      4.650,     46.905,      1.07,      2.795,     47.08, ...
## time (days) : 2016-08-22 to 2018-04-03
```

 Identifiez les informations suivantes concernant la série temporelle MODIS LST :

- système de projection et de coordonnées
- nombre et noms des attributs
- nombre total de “couches” temporelles
- dates minimum et maximum de la série temporelle
- unités des températures

 Sur le plot de la série temporelle MODIS LST, à quoi correspondent les carrés blancs (sans couleur) ? A votre avis, pourquoi n’y a-t-il pas de tels carrés blancs sur le plot de la série temporelle de précipitations ?

### 2.2.2 Construire les variables statistiques

Nos données météorologiques sont à présent disponibles. Pour constituer des modèles statistiques à partir de ces données, il faut en extraire des variables à “rattacher” aux tableau des données entomologiques. L’enjeu est de contruire des variables pertinentes au regard des connaissances sur l’impact des conditions météorologiques sur les moustiques.

Les conditions météorologiques telles que les températures et les précipitations impactent de nombreux traits de vie des vecteurs : émergence, croissance, survie, dispersion, activité, etc. Par exemple, la température affecte le moustique, à chaque étape de son cycle de vie (croissance larvaire, survie des adultes, etc.). De leur côté, les précipitations remplissent ou créent les gîtes larvaires et expliquent ainsi, en partie, la saisonnalité de l’abondance de certaines espèces d’anophèles. La fréquence des précipitation, leur abondance, leur durée, sont donc des paramètres essentiels pour expliquer les densités des vecteurs.

Pour cet exercice, nous allons donc créer les variables suivantes :

- températures minimum et maximum moyennes sur le mois précédent chaque collecte (1 mois = ~ durée de vie d’un anophele sur le terrain), dans une zone d’un rayon de 2 km autour des points de capture (2 km = ~ distance de vol maximum d’un anophele sur le terrain)
- cumul des précipitations selon ces mêmes paramètres

Ci-dessous, nous proposons un code pour réaliser ces opérations :

```
# créer une zone tampon d'un rayon de 2 km autour de chaque
# point de collecte
sp_buffer <- st_buffer(entomological_data, 2000)

# mapview(list(roi, entomological_data,
# sp_buffer), legend=F)

# écrire une fonction qui créer les variables statistiques,
# données en entrée : - une zone tampon (dans laquelle les
# données seront résumées), - une série temporelle de type
```

```

# SpatRaster, - une bande d'intérêt pour cette SpatRaster,
# - un intervalle de temps d'intérêt - une fonction pour
# résumer les données pour la période considérée

fun_get_zonal_stat <- function(sp_buffer, raster_ts, variable,
  min_date, max_date, fun_summarize) {

  r_sub <- terra::subset(raster_ts, terra::time(raster_ts) >=
    min_date & terra::time(raster_ts) <= max_date)
  r_agg <- terra::app(r_sub[variable], fun_summarize, na.rm = T)
  val <- terra::extract(r_agg, sp_buffer, fun = mean, ID = F,
    touches = TRUE, na.rm = T)
  val <- as.numeric(val)

  return(val)
}

# diviser l'ensemble de données (nécessaire pour
# l'exécution de la fonction)
sp_buffer_split <- split(sp_buffer, seq(nrow(sp_buffer)))

# exécuter la fonction pour obtenir les variables
# météorologiques souhaitées
LSTmax_1_month_bef <- purrr::map_dbl(sp_buffer_split, ~fun_get_zonal_stat(.,
  modis_ts, "LST_Day_1km", .$date - 30, .$date, "mean"))
LSTmin_1_month_bef <- purrr::map_dbl(sp_buffer_split, ~fun_get_zonal_stat(.,
  modis_ts, "LST_Night_1km", .$date - 30, .$date, "mean"))
rain_1_month_bef <- purrr::map_dbl(sp_buffer_split, ~fun_get_zonal_stat(.,
  gpm_ts, "precipitation", .$date - 30, .$date, "sum"))

# rattacher ces variables au tableau 'entomological_data'
entomological_data$LSTmax_1_month_bef <- LSTmax_1_month_bef -
  273.15 # le - 273.15 sert à convertir la température de Kelvin en °C
entomological_data$LSTmin_1_month_bef <- LSTmin_1_month_bef -
  273.15
entomological_data$rain_1_month_bef <- rain_1_month_bef

head(entomological_data)

## Simple feature collection with 6 features and 7 fields
## Geometry type: POINT
## Dimension: XY
## Bounding box: xmin: -5.614132 ymin: 9.18683 xmax: -5.514849 ymax: 9.288387
## Geodetic CRS: WGS 84
## mission date village n geometry LSTmax_1_month_bef
## 1 1 2016-09-21 GUE 120 POINT (-5.54897 9.271823) 28.59190
## 2 1 2016-09-21 LOK 156 POINT (-5.60539 9.18683) 27.31314
## 3 1 2016-09-21 PEN 84 POINT (-5.514849 9.253527) 28.50162
## 4 1 2016-09-22 KOL 131 POINT (-5.524137 9.288387) 26.71242
## 5 1 2016-09-23 KAG 165 POINT (-5.535534 9.282156) 27.68345
## 6 1 2016-09-23 NOK 188 POINT (-5.614132 9.251798) 29.66437
## LSTmin_1_month_bef rain_1_month_bef
## 1 20.11336 377.1600

```

## 2	19.89634	366.2787
## 3	18.71333	374.3600
## 4	20.03546	362.1925
## 5	20.29548	370.0350
## 6	19.92835	368.2850

Nos variables météorologiques sont prêtes ! Place au traitement des données paysagères.

## 2.3 Importer et préparer les données paysagères

Pour les données paysagères, nous allons utiliser une carte d'occupation du sol. Il existe de nombreuses cartes d'occupation du sol en libre accès, à des résolutions spatiales qui diffèrent. Ici, nous allons utiliser une carte d'occupation du sol à l'échelle du continent africain produite par l'Agence Spatiale Européenne en 2016 à partir d'images satellitaires Sentinel 2. La produit est libre et accessible à l'adresse suivante : <https://2016africalandcover20m.esrin.esa.int/viewer.php>.

### 2.3.1 Importer les données

Comme la carte à l'échelle de l'Afrique entière est très volumineuse (plus de 5 GB), nous l'avons téléchargée en amont de ce TD et pré-découpée sur notre zone d'étude. La carte est ainsi disponible dans le dossier `data/landscape_data/landcover.tif`.

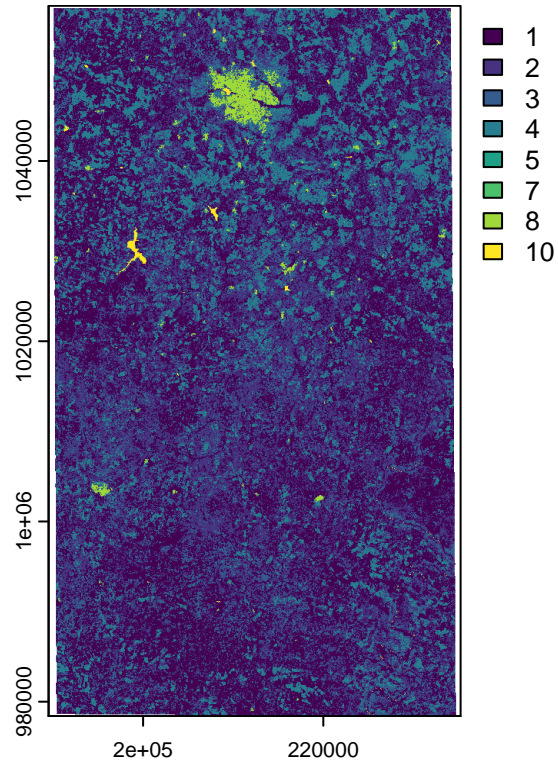
Chargeons la dans R à l'aide de la librairie R `terra` :

```
# importer les données d'occupation du sol
raster_lulc <- terra::rast("data/landscape_data/landcover.tif")

raster_lulc
```

```
## class      : SpatRaster
## dimensions  : 3874, 2229, 1  (nrow, ncol, nlyr)
## resolution  : 20.51034, 20.34897  (x, y)
## extent     : 189514.3, 235231.8, 978391.3, 1057223  (xmin, xmax, ymin, ymax)
## coord. ref. : WGS 84 / UTM zone 30N (EPSG:32630)
## source     : landcover.tif
## name       : landcover
```

```
terra::plot(raster_lulc)
```



🔧 Identifiez les informations suivantes concernant le raster d'occupation des sols :

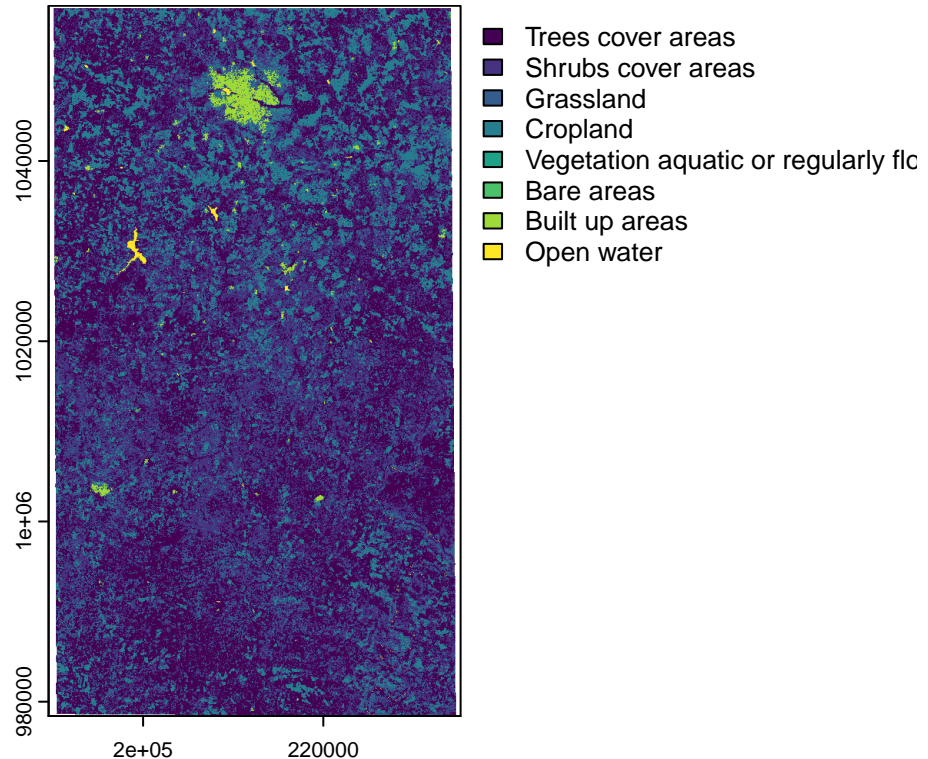
- résolution spatiale précise
- système de projection et de coordonnées
- nombre total de pixels

Pour l'instant, nous ne savons pas à quelle classe d'occupation du sol correspondent les valeurs des pixels. Ces informations se trouvent dans le fichier disponible sous `data/landscape_data/landcover_rat.csv`. Chargeons cette table attributive du raster (*raster attribute table*) afin d'obtenir la signification des pixels :

```
rat <- read.csv("data/landscape_data/landcover_rat.csv")

levels(raster_lulc) <- rat

terra::plot(raster_lulc)
```



Enfin, superposons la carte d'occupation des sols et les villages de collecte des anophèles. Pour cela, nous devons convertir le système de coordonnées de la couche géographique des données entomologiques de sorte qu'elle corresponde à celui de la couche d'occupation du sol :

```
entomological_data_utm <- sf::st_transform(entomological_data,
  terra::crs(raster_lulc))

# terra::points(terra::vect(entomological_data_utm),
# pch=21, col='black', bg='white', cex=1)
```

### 2.3.2 Construire les variables statistiques

Comme pour les données météorologiques, il s'agit maintenant de constituer des variables statistiques à partir des données paysagères. Pour ceci, nous allons extraire un certain nombre de métriques paysagères (en anglais : *landscape metrics*) dans des zones tampons autour des points de collecte de moustiques. Il existe, dans l'absolu, un grand nombre de métriques paysagères. Dans cet exercice, nous allons en calculer une simple : le pourcentage de surface utilisé par chaque classe d'occupation du sol dans une zone tampon d'un rayon de 2 km autour de chaque village.

Afin de calculer ces métriques, nous allons utiliser la librairie R `landscapemetrics`.

Ci-dessous, nous proposons un code pour réaliser ces opérations :

```

library(landscapemetrics)
library(tidyr)

villages <- unique(entomological_data_utm[c("village", "geometry")])

# La fonction qui suit permet de calculer le % de surface
# utilisé par chaque classe d'occupation du sol dans une
# zone tampon d'un rayon de 2 km autour de chaque village.
# Pour davantage de détail sur la fonction, tapez :
# help(sample_lsm)

df_lsm <- landscapemetrics::sample_lsm(landscape = raster_lulc,
  y = villages, plot_id = villages$village, what = "lsm_c_pland",
  shape = "circle", size = 2000, verbose = F)

# Afin d'obtenir le nom des classes d'occupation du sol (et
# pas seulement le numéro des pixels), joignons la table
# attributaire du raster
rat$pixlabel <- gsub(" ", "_", rat$pixlabel)
df_lsm <- dplyr::left_join(df_lsm, rat, by = c(class = "pixval"))

# Passons le tableau des métriques paysagères du format
# 'long' au format 'large'
df_lsm <- df_lsm %>%
  dplyr::select(value, plot_id, pixlabel) %>%
  tidyr::pivot_wider(names_from = pixlabel, values_from = value,
    values_fill = 0)

# Puis joignons les tableaux de capture des anophèles à
# celui des métriques paysagères
entomological_data_utm <- dplyr::left_join(entomological_data_utm,
  df_lsm, by = c(village = "plot_id"))
entomological_data_utm <- st_drop_geometry(entomological_data_utm)

head(entomological_data_utm)

```


```

## mission      date village  n LSTmax_1_month_bef LSTmin_1_month_bef
## 1          1 2016-09-21   GUE 120          28.59190          20.11336
## 2          1 2016-09-21   LOK 156          27.31314          19.89634
## 3          1 2016-09-21   PEN  84          28.50162          18.71333
## 4          1 2016-09-22   KOL 131          26.71242          20.03546
## 5          1 2016-09-23   KAG 165          27.68345          20.29548
## 6          1 2016-09-23   NOK 188          29.66437          19.92835
## rain_1_month_bef Trees_cover_areas Shrubs_cover_areas Grassland Cropland
## 1          377.1600          28.76573          38.91389 11.935346 19.77397
## 2          366.2787          23.28241          45.46410 20.542139 10.71135
## 3          374.3600          42.48671          32.70859  8.090987 16.71371
## 4          362.1925          41.12374          32.41166  8.620123 17.61789
## 5          370.0350          36.63688          27.99750 11.408580 23.34604
## 6          368.2850          22.57704          45.16723  7.700900 23.55608
## Built_up_areas Open_water Vegetation_aquatic_or_regularly_flooded Bare_areas
## 1          0.6044877          0          0.006570518 0.000000000

```



## 2	0.0000000	0	0.000000000	0.000000000
## 3	0.0000000	0	0.000000000	0.000000000
## 4	0.2134507	0	0.006567713	0.006567713
## 5	0.6044281	0	0.006569871	0.000000000
## 6	0.9921808	0	0.000000000	0.006570734

 Exprimez en langage naturel (i.e. en français) l'information que fournit la première ligne du tableau.

### 3 Modéliser

Nous avons donc construit un tableau contenant des **variables dépendantes** (aussi appelées ‘variables à expliquer’ ou ‘variables à prédire’) et des **variables indépendantes** (aussi appelées ‘variables explicatives’ ou ‘variables prédictives’). Ce tableau va nous permettre de modéliser les abondances des anophèles.

Mais en fait, qu’est ce que la modélisation statistique, et à quoi sert elle ? Tout un sujet ... mais pour résumer, un modèle statistique est un outil permettant d’associer des données, à savoir des informations mesurables du monde qui nous entoure (ou *observations*). Associer des données peut servir différents enjeux scientifiques :

- **expliquer** : tester ou vérifier une théorie scientifique,
- **explorer** : mieux comprendre un phénomène d’intérêt,
- **prédire** : prédire de nouvelles valeurs d’un événement.

Selon l’enjeu poursuivi, l’ensemble du processus de modélisation statistique diffère : choix des variables, choix des modèles, méthodes d’évaluation des modèles, etc.

Si vous souhaitez en savoir plus sur les enjeux de la modélisation stastique, voici 2 lectures intéressantes :

- Tredennick, A. T., Hooker, G., Ellner, S. P., & Adler, P. B. (2021). *A practical guide to selecting models for exploration, inference, and prediction in ecology*. In Ecology (Vol. 102, Issue 6). Wiley. <https://doi.org/10.1002/ecy.3336>
- Paul Taconet. *Fouille de données spatio-temporelles pour l’étude du risque de transmission résiduelle du paludisme à échelle paysagère en milieu rural ouest-africain* (Chapitre 2 : *Étude des systèmes complexes et modélisation statistique*). Médecine humaine et pathologie. Université de Montpellier, 2022. Français. <https://theses.hal.science/tel-03841709>

Dans cet exercice, nous allons générer des modèles de l’abondance des moustiques pour ces 3 enjeux (explicatif, exploratoire, prédictif). L’objectif pédagogique est ici de montrer comment, à partir du tableau que nous avons généré dans la partie précédente, nous pouvons répondre à différentes questions de recherche grâce à la modélisation statistique. Les formes de modélisation et questions de recherche associées sont présentées ci-dessous :

- Modélisation explicative : **Quel est l’impact précis de certaines variables environnementales (températures, précipitations) sur les densités agressives ?**
- Modélisation exploratoire : **Quels sont les déterminants environnementaux des abondances des moustiques ?**
- Modélisation prédictive : **Est-il possible de prédire les abondances des moustiques dans d’autres villages de la zone d’étude grâce aux données environnementales ?**

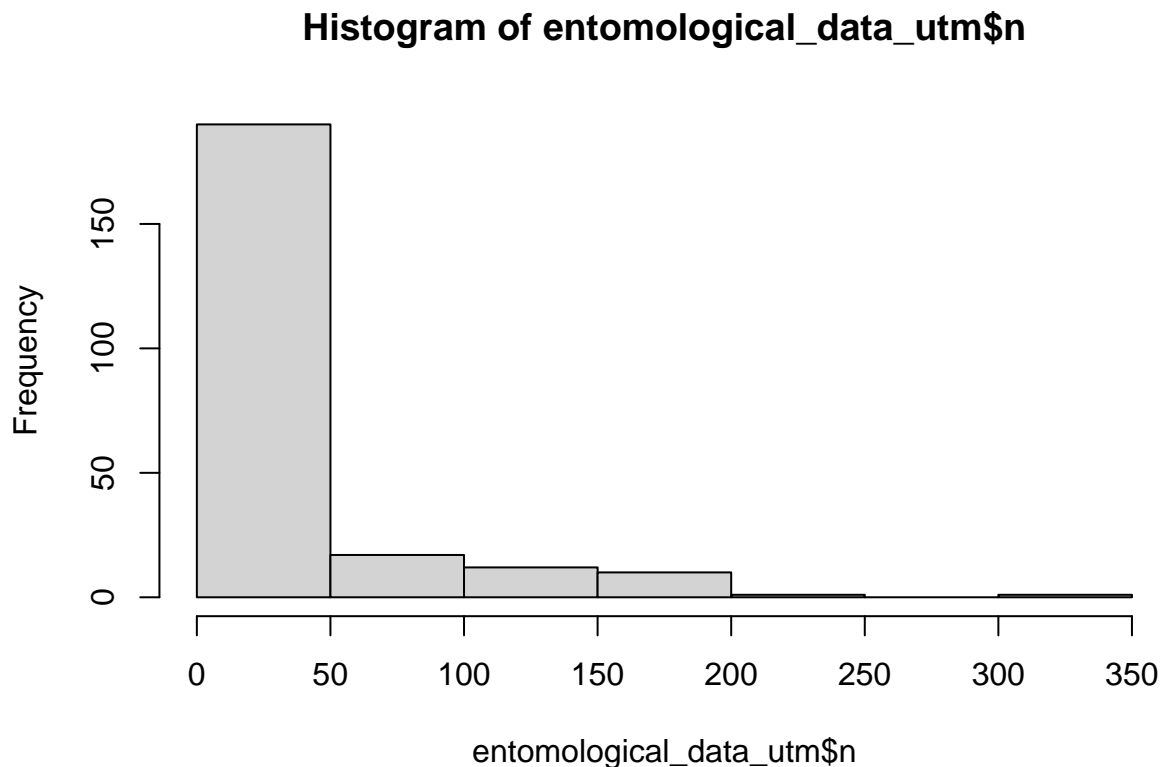
C’est parti !

### 3.1 Visualiser les données et leurs associations

Quelle que soit l'enjeu et la question de recherche, tout travail de modélisation commence par la visualisation des données sur des graphiques pertinents.

Ici, nous allons tout d'abord visualiser la distribution statistique de la variable dépendante (l'abondance des moustiques) :

```
hist(entomological_data_utm$n)
```



🔧 Comment qualifieriez-vous cette distribution statistique ? Qu'a-t-elle de particulier ?

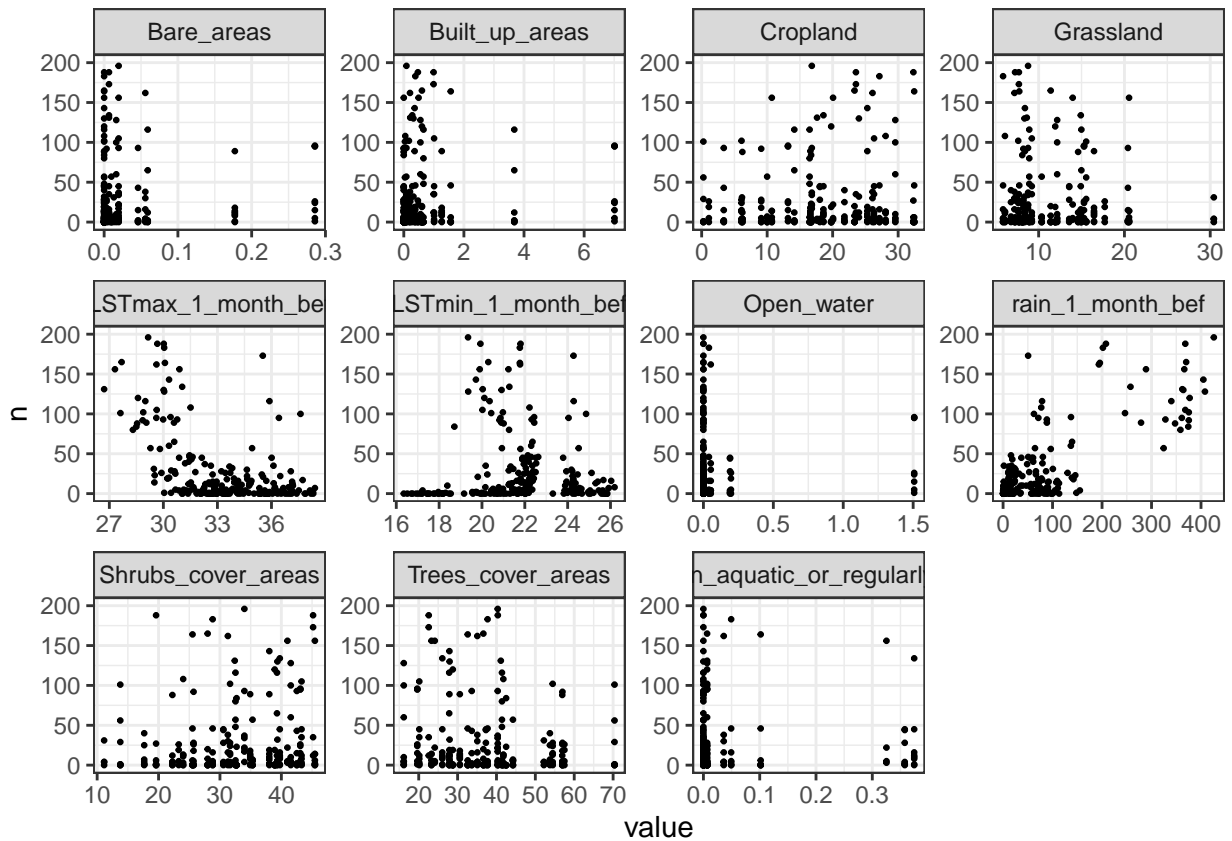
Nous allons également visualiser la forme des associations entre la variable dépendante (abondance des moustiques) et l'ensemble des variables indépendantes, grâce des des graphiques en nuage de points. Pour cela, nous utilisons la librairie R `ggplot2` :

```
library(ggplot2)

viz <- entomological_data_utm %>%
  tidyr::pivot_longer(LSTmax_1_month_bef:Bare_areas, names_to = "variable",
    values_to = "value") %>%
  ggplot2::ggplot(aes(x = value, y = n)) + geom_point(size = 0.5) +
  facet_wrap(. ~ variable, scales = "free") + ylim(c(0, 200)) +
```

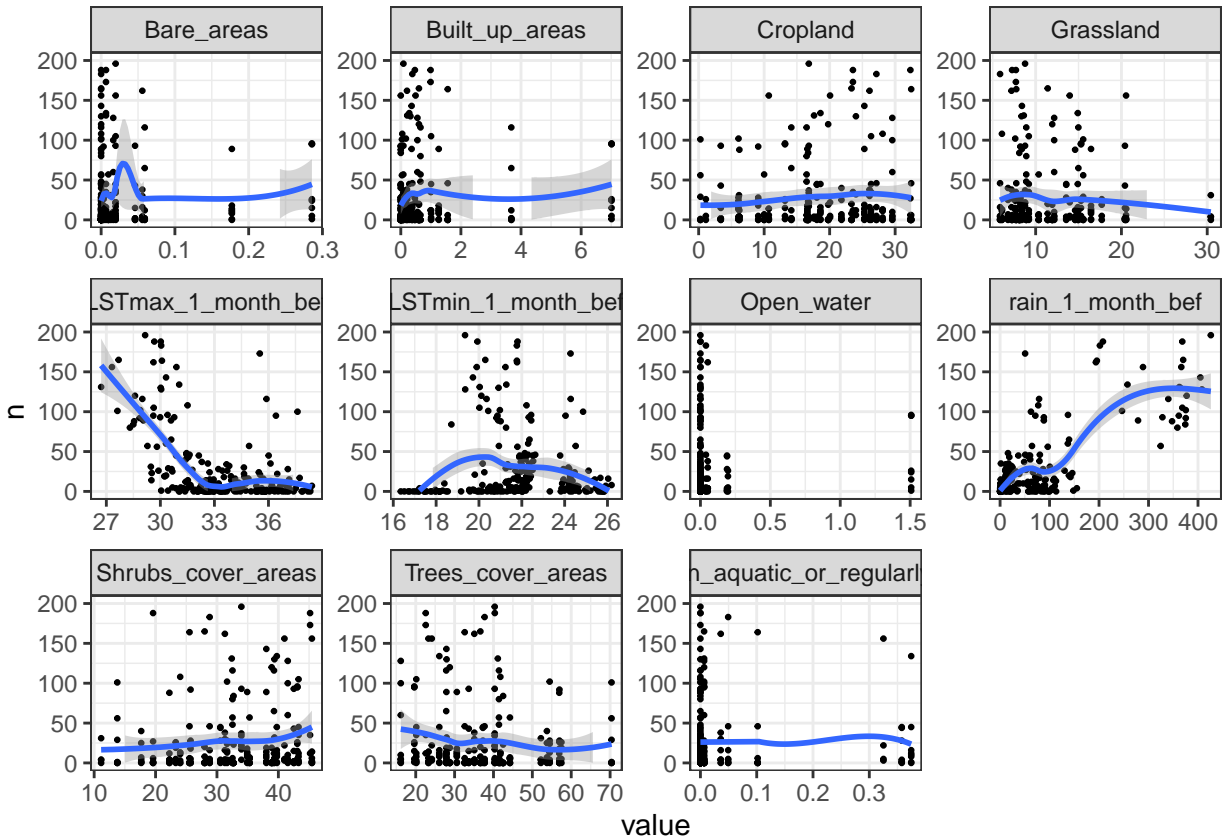
```
theme_bw()
```

```
viz
```



La nature des relations entre les variables peut être délicate à visualiser sur ce genre de graphique. Ajoutons une courbe de regression qui aidera à visualiser :

```
viz + geom_smooth()
```



🔧 Selon les données, avec quelles variables d'occupation du sol ou de météorologie l'abondance des anophèles est-elle positivement corrélée ? À l'inverse, avec quelles variables est-elle négativement corrélée ?

🔧 Quelles sont les variables qui ont l'air d'impacter le plus l'abondance des anophèles ?

### 3.2 Modélisation explicative

Ici, nous allons utiliser la modélisation statistique afin de répondre à la question suivante : **Quel est l'impact précis de certaines variables environnementales (températures, précipitations) sur les densités agressives ?** Quantifier précisément l'association entre des variables statistiques requiert d'utiliser des modèles transparents, qui délivrent des informations chiffrées sur le sens, la force, et la significativité de l'association. Les modèles paramétriques type modèles linéaires sont parfaitement adaptés à ce genre de situations.

Ici, nous allons donc effectuer une régression linéaire en ajustant un modèle linéaire. Pour cela nous allons utiliser la librairie R MASS.

Comme la distribution de la variable réponse est binomiale négative, nous utilisons la famille `nbinom2` dans l'argument `family` de la fonction `glmTMB`.

Ajustons un modèle linéaire entre le nombre d'anophèles collectés et le cumul des précipitations sur le mois précédant la collecte :

```
library(glmTMB)

df_explanatory_model <- entomological_data_utm

mod1 <- glmTMB(n ~ rain_1_month_bef, data = entomological_data_utm,
  family = nbinom2)
summary(mod1)
```

```
## Family: nbinom2 ( log )
## Formula:          n ~ rain_1_month_bef
## Data: entomological_data_utm
##
##      AIC      BIC   logLik deviance df.resid
##  1723.0   1733.3   -858.5   1717.0      228
##
##
## Dispersion parameter for nbinom2 family (): 0.452
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.107436   0.134141  15.711  <2e-16 ***
## rain_1_month_bef 0.009775   0.001190   8.216  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

 Exprimez en français les informations que fournit le modèle.

 Générez une regression linéaire avec la variable 'LSTmax\_1\_month\_bef' et la variable 'Cropland' puis interprétez les résultats.

### 3.3 Modélisation exploratoire

Ici, nous allons utiliser la modélisation statistique afin de répondre à la question suivante : **Quels sont les déterminants environnementaux des abondances des moustiques ?** Pour ce faire, nous avons besoin de modèles capables de capturer de manière autonome des associations complexes entre les variables (non-linéarités, interactions), potentiellement non-hypothétisées. Tout l'intérêt et l'enjeu résidera ensuite dans l'interprétation de ces modèles, afin de révéler les associations qu'ils ont 'appprises' ou capturées. L'interprétation de ces associations, à la lumière des connaissances actuelles, nous permettra d'émettre des hypothèses sur les principaux déterminants environnementaux des abondances des moustiques. Les modèles non-paramétriques de *machine learning* type forêts aléatoires sont adaptés à ce genre de situations.

Le travail de modélisation exploratoire se déroule en deux phases :

- identification et suppression des variables collinéaires (la présence de variables collinéaires dans un modèle multivarié peut fausser les résultats)
- ajustement d'un modèle de *machine learning*
- interprétation des associations que le modèle a capturées grâce à des outils d'interprétation de ce modèles (outils dits *interpretable machine learning*).

Trois librairies R seront nécessaires : `randomForest`, `CAST`, `caret` .

```

library(randomForest)
library(CAST)
library(caret)

df_exploratory_model <- entomological_data_utm

# définissons les variables indépendantes
predictors <- colnames(df_exploratory_model[5:ncol(df_exploratory_model)])

# identifions les variables indépendantes qui sont
# corrélées au dessus de 0.7 (coeff de corrélation de
# pearson) :
m <- cor(df_exploratory_model[, predictors], method = "pearson",
        use = "pairwise.complete.obs")
index <- which(abs(m) > 0.7 & abs(m) < 1, arr.ind = T)
df_cor <- subset(as.data.frame(index), row <= col)
p <- cbind.data.frame(stock1 = rownames(m)[df_cor[, 1]], stock2 = colnames(m)[df_cor[,
2]])

p

```

```

##           stock1           stock2
## 1 Trees_cover_areas Shrubs_cover_areas
## 2   Built_up_areas   Open_water
## 3   Built_up_areas   Bare_areas
## 4     Open_water   Bare_areas

```

Les trois variables suivantes sont corrélées : Bare\_areas / Open\_water / Built\_up\_areas ; et les deux variables suivantes sont corrélées : Trees\_cover\_areas / Shrubs\_cover\_areas.

Nous devons retirer certaines de ces variables afin d'ôter le problème de multicollinéarité.

```

predictors <- setdiff(predictors, c("Shrubs_cover_areas", "Bare_areas",
"Open_water"))

```

Ajustons à présent le modèle de forêts aléatoires aux données.

```

indices_cv <- CAST::CreateSpacetimeFolds(df_exploratory_model,
    spacevar = "village", k = length(unique(df_exploratory_model$village)))

tr = caret::trainControl(method = "cv", index = indices_cv$index,
    indexOut = indices_cv$indexOut, savePredictions = "final")

mod <- caret::train(x = df_exploratory_model[, predictors], y = df_exploratory_model$n,
    method = "rf", tuneLength = 10, trControl = tr, metric = "MAE",
    maximize = FALSE, preProcess = c("center", "scale"))

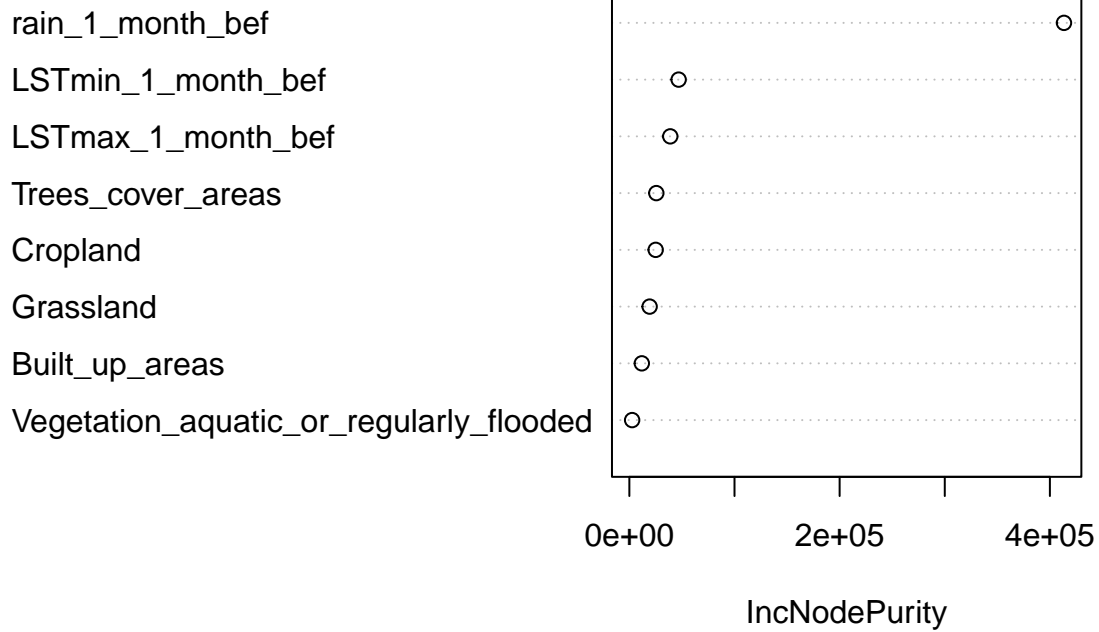
```

## note: only 7 unique complexity parameters in default grid. Truncating the grid to 7 .

Enfin, interprétons le modèle à l'aide de deux outils : l'importance des variables et les 'partial dependence plots' :

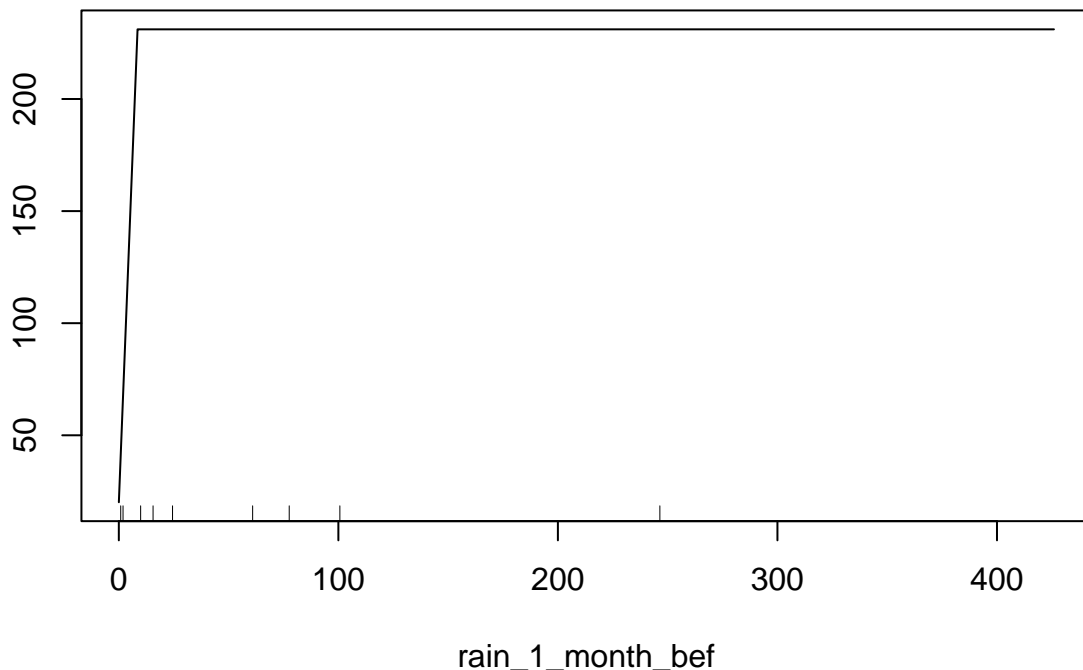
```
# Importance des variables
randomForest::varImpPlot(mod$finalModel, main = "Variable importance plot")
```

### Variable importance plot



```
## Partial dependence plot pour la variable
## 'rain_1_month_bef'
partialPlot(mod$finalModel, df_exploratory_model, rain_1_month_bef)
```

### Partial Dependence on rain\_1\_month\_bef



### 3.4 Modélisation prédictive

Ici, nous allons utiliser la modélisation statistique afin de répondre à la question suivante : **Est-il possible de prédire les abondances des moustiques dans d'autres villages de la zone d'étude grâce aux données environnementales ?** Pour ce faire, nous avons besoin de modèles capables de prédire au mieux. Les modèles non-paramétriques de *machine learning* type forêts aléatoires sont, là aussi, comme pour la modélisation exploratoire, adaptés à ce genre de situations.

La chaîne de traitement pour ajuster le modèle ressemble fortement à celle de la modélisation exploratoire, à quelques différences prêts :

- en modélisation prédictive, l'enjeu n'étant pas l'interprétation des modèles, la multicollinéarité n'est pas un problème (sauf si le nombre de variables indépendantes est très important)
- une fois le modèle généré, l'enjeu ici n'est pas de l'interpréter mais d'évaluer sa puissance prédictive, c'est à dire sa capacité à prédire les abondances des moustiques dans de nouveaux villages ou à de nouvelles dates (i.e. des villages/dates qui n'ont pas servi pour l'entraînement du modèle)

Pour commencer, ajustons le modèle de forêts aléatoire en utilisant toutes les variables indépendantes :

```
library(randomForest)
library(CAST)
library(caret)

df_predictive_model <- entomological_data_utm
```



```

predictors <- colnames(df_exploratory_model[5:ncol(df_exploratory_model)])

indices_cv <- CAST::CreateSpacetimeFolds(df_predictive_model,
  spacevar = "village", k = length(unique(df_predictive_model$village)))

tr = caret::trainControl(method = "cv", index = indices_cv$index,
  indexOut = indices_cv$indexOut, savePredictions = "final")

mod <- caret::train(x = df_predictive_model[, predictors], y = df_predictive_model$n,
  method = "rf", tuneLength = 10, trControl = tr, metric = "MAE",
  maximize = FALSE, preProcess = c("center", "scale"))

```

Evaluons maintenant la puissance prédictive du modèle :

```

df_predictive_model$rowIndex <- seq(1, nrow(df_predictive_model),
  1)

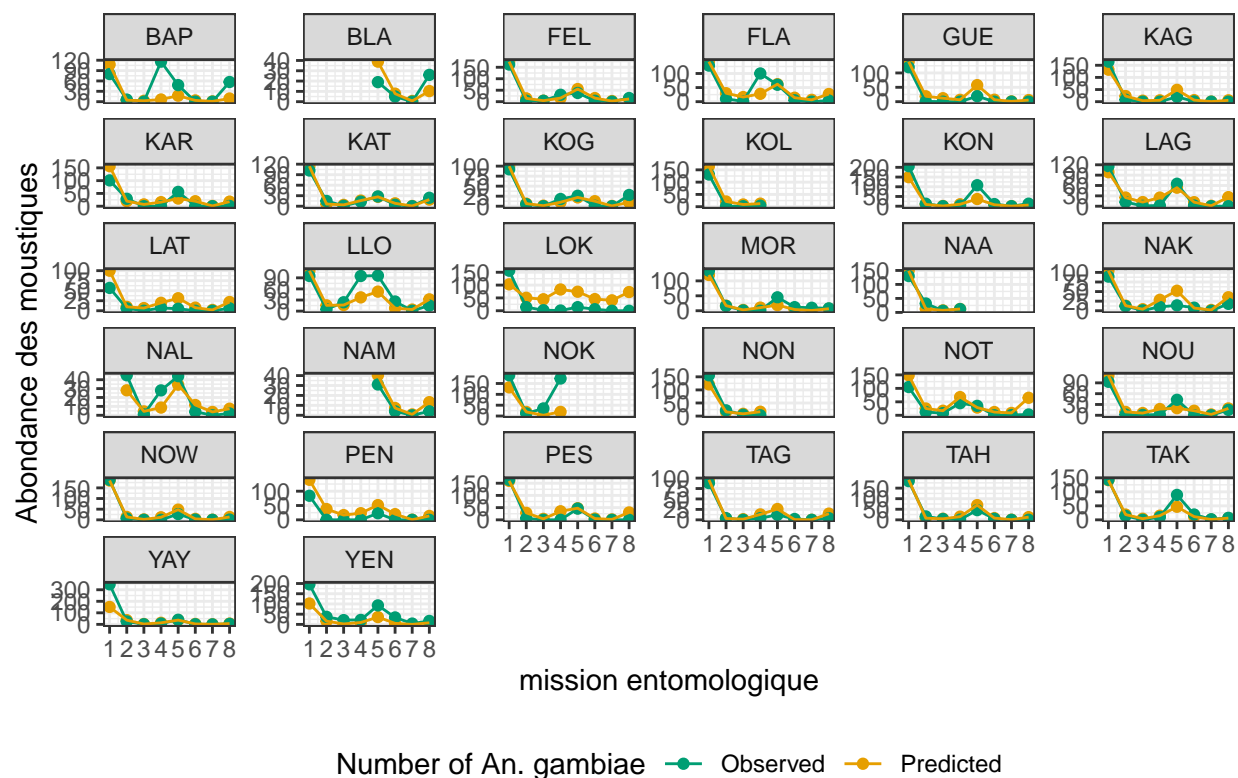
df_cv <- mod$pred %>%
  left_join(df_predictive_model) %>%
  dplyr::select(pred, obs, mission, village)

plot_eval_abundance_model <- df_cv %>%
  dplyr::group_by(mission, village) %>%
  dplyr::summarise(pred = mean(pred), obs = mean(obs)) %>%
  as_tibble() %>%
  pivot_longer(c("pred", "obs")) %>%
  mutate(name = ifelse(name == "pred", "Predicted", "Observed")) %>%
  ggplot(aes(x = mission, y = value, color = name)) + geom_point() +
  geom_line() + facet_wrap(. ~ village, scales = "free_y") +
  theme_bw() + scale_colour_manual(values = c("#009E73", "#E69F00"),
  na.translate = F) + scale_x_continuous(breaks = c(1, 2, 3,
  4, 5, 6, 7, 8)) + xlab("mission entomologique") + ylab("Abondance des moustiques") +
  labs(color = "Number of An. gambiae") + theme(legend.position = "bottom") +
  ggtitle("Valeur observée vs. prédite par village et mission entomologique")

plot_eval_abundance_model

```

## Valeur observée vs. prédite par village et mission entomologique



Le modèle semble prédire correctement !

## 4 Pour aller plus loin

-> fouille de données

### License

Cette œuvre est mise à disposition selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale 4.0 International](https://creativecommons.org/licenses/by/4.0/).

### Citation

Paul Taconet. (2023, février 13). TD SIG niveau avancé - Modélisation des dynamiques spatio-temporelles des abondances des vecteurs du paludisme au Burkina Faso avec le langage de programmation R. Zenodo. <https://doi.org/10.5281/zenodo.7635937>