readxl::read\_excel("Data/Récap projet EDGG\_A.xlsx", (Lire le fichier excel dans le dossier Data et qui s’appelle récap…xlsx)

sheet = "PLAN DE PLAQUES", (Je choisis e

nsuite la feuille qui s’appelle PLAN DE PLAQUES)

col\_names = FALSE) %>%

select(1:10) %>% (Je sélectionne les colonnes de 1 à 10 = Les colonnes qui contient mes plan de plaques en couleurs comme le fait Charlotte)

fill(1, .direction = c("down")) %>% (Prendre ce qui est inscrit sur la première colonne et tu le copies dans les lignes en bas, quand on arrive à une nouvelle inscription on commence à copier la nouvelles dans les lignes en dessous = Dans notre cas « Plaque EDGGA\_1 » s’inscrit pour les 12 lignes de la palque et ça continue pour les 4 plaques en changeant uniquement le dernier chiffre par 2, 3 et 4

row\_to\_names(row\_number = 1) %>% (Nommer les colonnes par ce qui est inscrit dans la première ligne

rename(plate\_row = 2, plate = 1) %>% (renommer ce qui était Plaque EDGGA\_1 par plate = pemière colonne, et renommer ce qui était NA par plate\_row = deuxième colonne)

pivot\_longer(

cols = -c(plate, plate\_row), (pivot longer pour changer la direction du tableau, Les colonnes à pivoter sont plate et plate row)

names\_to = "plate\_column", (Une nouvelle colonne qui s’appelle plate\_column)

values\_to = "Sample code" (Une nouvelle colonne qui s’appelle Samplecode et qui prend les valeurs croisés entre plate row et plate column)

) %>%

filter(plate\_row != 0) %>% (j’élimine les lignes de plate row qui ont des valeurs égales à zero, ils doivent contenir des valeurs entre 1 et 12 ce qui correspond aux noms des colonnes de mes plaques)

mutate( (me permet de modifier le colonnes)

plate = str\_remove(plate, "PLAQUE "), (Dans la colonne plate, je supprime PLAQUE et l’espace derrière comme ça il me reste que EDGGA\_1

plate = str\_replace(plate, "\_", "-"), (Dans plate je remple les tirets 8 par tirets 6)

plate\_row = as.character(plate\_row), (render la colonne plate\_row comme étant character)

plate\_row = str\_pad(

plate\_row,

width = 2,

side = c("left"),

pad = "0" (Dans plate row je vectorise les valeurs de 1 à 12, cad : je veux que qu’il prend un width de deux cvd il doit avoir deux chiffres et s’il n’a pas deux chiffres il doit « pad cad coller un zero au left pour ainsi avoir 01 au lieu de 0)

)

) %>%

unite(plate, plate, plate\_column, sep = "\_") %>% (permet d’unir dans plate les inscriptions de plate + plate column et les séparer par un tiret 8)

unite(plate, plate, plate\_row, sep = "") %>% (mnt que j’ai plate : EDGGA-1\_A, il me faut le numéro de la ligne, donc je colle dans la plate, le nom de plate le nouveau + plate\_row et je ne les sépare pas donc collés ça donne EDGGA-1\_A12

rename(`Sequencing plate` = plate) -> metadata\_for\_Denis (Je renomme plate par sequencing plate, puis j’enregistre le fichier comme étant metadata\_for\_Denis

metadata\_for\_Denis %>% head()

#Add "Host community" from sheet "EDGG1" in file "Echantillonnage EDGG 21\_CAM.xls"----

readxl::read\_excel("Data/Echantillonnage EDGG 21\_CAM.xlsx",

sheet = "EDGG1",

col\_names = FALSE,

n\_max = 1) %>%

pull() %>%

str\_remove(".\* ") -> sample\_code\_prefix

readxl::read\_excel(

"Data/Echantillonnage EDGG 21\_CAM.xlsx", (je lis mon fichier EDGG\_21\_CAM, je sélectionne la feuille qui s’appelle EDGG1, je sélectionne uniquement à partir de la 9 èm ligne

sheet = "EDGG1",

col\_names = TRUE,

skip = 8

) %>%

rename(plants = 1) %>% (je nomme la première colonne de la première ligne par plants)

pivot\_longer(cols = -plants, (On pivote ensuite notre tableau en gardant la colonne plants, on ajoute une colonne qui s’appelle sample code names to c’est toujours la deuxième colonne dans pivot longer, ensuite une troisième colonne qui prend le nom cover et qui prend les valeurs en croisant les infos de plants et sample code

names\_to = "Sample code",

values\_to = "cover",

values\_drop\_na = TRUE) %>% (sert à éliminer les lignes qui contiennent des valeurs NA)

filter(!plants %in% c("VÃ©gÃ©tation", "Sol nu", "LitiÃ¨re")) %>% b(sert à éliminer dans plants tout ce qui s’appelle végétation, sol nu et litière

mutate(cover = str\_remove(cover, "%"), (permet d’éliminer pourcentage dans cover)

cover\_classes = case\_when( 3.

(permet de rempacer les fouchettes de cover par un seul nombre)

cover == "1-5" ~ "3",

cover == "<1" ~ "0,5",

cover == "5-15" ~ "10",

cover == "15-25" ~ "20",

cover == "25-50" ~ "37,5",

cover == "50-75" ~ "62,5",

cover == ">75" ~ "87,5",

TRUE ~ "oups"

)

) %>%

select(-cover) %>% (On sélectionne les valeurs de cover)

unite(host\_community, plants, cover\_classes, sep = "\_") %>% (Puis on uni dans une colonne host\_community ce qui est inscrit dans plants puis on colle juste après ce qu’on a dans cover\_classes et on sépare les deux infos par un tiret 8

group\_by(`Sample code`) %>% (je groupe par sample\_code)

summarise(host\_community = paste0(host\_community, collapse = ";")) %>%

ungroup() %>% (permet de coller tout ce qu’il y a dans host community et les mettre les unes après les autres avec une séparation ; et ce pour chaque échantillon)

mutate(`Sample code` = str\_remove(`Sample code`, ".\* "), (il faut mnt changer le sample code parce que ici il s’appelle quadrat 1 moi je veux l’appeler avec grid name plus les deux chiffres derrière, donc je commence par supprimer tout ce qui est avant le dernier espace il me reste uste le chiffre 1)

`Sample code` = str\_pad(`Sample code`, width = 2, (ensuite je pad cad que je veux que mon chiffre se compose de deux chiffres)

side = c("left"), (je colle à la gauche du chiffre 1)

pad = "0"), (je colle le chiffre zero)

`Sample code` = paste0(sample\_code\_prefix, `Sample code`), (Dans la colonne Sample code, je colle Grid\_name par exemple 21\_CAM\_01, et je colle juste après mon sample code qui se compose mnt de deux chiffres)

`Sample code` = paste0(`Sample code`, "00")) -> other\_metadata mnt que j’ai mon sample code à 4 chiffres moi je le veux à six chiffres, les deux derniers zero étant le numero du replica

#Add date----

readxl::read\_xlsx("Data/Echantillonnage EDGG 21\_CAM.xlsx",

sheet = "EDGG1",

col\_names = FALSE,

skip = 1,

n\_max = 1) %>%

pull(`...1`) %>%

str\_remove(".\*: ") %>%

as.Date(format="%d/%m/%Y") -> date

other\_metadata %>%

mutate(`Collection date`=date)->other\_metadata

#Add GPS----

readxl::read\_xlsx("Data/Echantillonnage EDGG 21\_CAM.xlsx",

sheet = "EDGG1",

col\_names = FALSE,

skip = 3,

n\_max = 1) %>%

pull(`...1`) %>%

str\_remove(".\*sur ") -> which

#Assemble everything----

metadata\_for\_Denis <- left\_join(x=metadata\_for\_Denis, y=other\_metadata,by=c("Sample code"))

* Deuxième script : Distribution des abondances relatives

#EXPLORING THE PLAN DATASET----

#install.packages("janitor")

#Load packages----

library(readxl)

library(tidyverse)

library(janitor)

#Testing bits of the script----

fichier\_test<-"Data/Echantillonnage EDGG 21\_CAM.xlsx" (Dans un fichier test on met le fichier echantillonnage cam 2021 du dossier Data de notre directory)

feuille\_test<-"EDGG1" (On prend uniquement la première feuille qui s’appelle EDGG1)

fichier\_test %>%

str\_remove(., ".\* ") %>%

str\_remove(., ".xlsx") -> grid\_name\_base

feuille\_test %>%

str\_remove(., "EDGG") %>%

str\_pad(

.,

width = 2,

side = c("left"),

pad = "0"

) -> grid\_number (déjà vu)

readxl::read\_excel(

fichier\_test,

sheet = feuille\_test,

col\_names = TRUE,

skip = 8 (On lit le fichier test puis on sélectionne uniquement la première page qu’on a nommé feuille\_test ensuite on saute toutes les lignes et on commence à lire à partir de la 9èm ligne)

) %>%

rename(Quadrat = 1) %>% (On renomme la première colonne de la première ligne, ici la 9ème, Quadrat

filter(Quadrat == "Végétation") %>% (à partir de la colonne quadrat on sélectionne uniquement ce qu’on appelle végétation)

pivot\_longer(cols =everything(), On fait ensuite un Pivot longer de, la première colonne étant quadrat, la deuxième colonne étant végétation à laquelle on donne les valeurs de végétation aux différents quadrats, pou là où y a pas de valeurs c’est vrai c’est égal à zero ou NA)

names\_to = "Quadrat",

values\_to = "Vegetation",

values\_drop\_na = TRUE)

grid\_name = paste(grid\_name\_base, grid\_number, sep = "\_") (Pour avoir le nom de la grille en total)

readxl::excel\_sheets(fichier\_test) %>%

subset(., str\_starts(string=.,pattern="EDGG")) -> sheets (Permet de lire uniquement les feuilles qui commencent par EDGG)

#THE function to read a grid (sheet) in a place and year (file)----

read\_a\_given\_sheet<-function(filename, sheet\_name){

#Code à  6 chiffres année\_localité

filename %>%

str\_remove(., ".\* ") %>%

str\_remove(., ".xlsx") -> grid\_name\_base

#Numéro de la grille

sheet\_name %>%

str\_remove(., "EDGG") %>%

str\_pad(

.,

width = 2,

side = c("left"),

pad = "0") -> grid\_number

#Indice de végétation pour chaque quadrat

readxl::read\_excel(

filename,

sheet = sheet\_name,

col\_names = TRUE,

skip = 8

) %>%

rename(Quadrat = 1) %>%

filter(Quadrat == "Végétation") %>%

pivot\_longer(cols =everything(),

names\_to = "Quadrat",

values\_to = "Vegetation", (Au-dessus, tout est déjà vu)

values\_drop\_na = TRUE) -> Vegetation\_by\_quadrat (On nomme ce tableau à deux colonnes « vgetation\_by\_quadrat)

#Indice de végétation pour chaque quadrat

readxl::read\_excel(

filename,

sheet = sheet\_name,

col\_names = TRUE,

skip = 8

) %>%

rename(Plant = 1) %>% (On crée un deuxième tableau en nomment la première colonne Plant

pivot\_longer(cols = -Plant, (ensuite une deuxième colonne qui s’appelle quadrat et qui prend les valeurs de cover

names\_to = "Quadrat",

values\_to = "Cover",

values\_drop\_na = TRUE) %>%

filter(!Plant %in% c("Végétation", "Sol nu", "Litière")) %>% (On supprime les trois lignes végétation, litière et sol nu)

mutate(Year = str\_sub(grid\_name\_base, 0,2), (sortir l’année d’échantillonnage)

Locality = str\_sub(grid\_name\_base, 4,6), (la localité d’échantillonnage)

Grid = paste(grid\_name\_base, grid\_number, sep = "\_"), (Le nom de la grille)

Cover = str\_remove\_all(Cover, "%"), (On enlève les pourcentages de toutes les fourchettes de recouvrement)

Cover\_classes = case\_when(

Cover == "1-5" ~ "3",

Cover == "<1" ~ "0.5",

Cover == "5-15" ~ "10",

Cover == "15-25" ~ "20",

Cover == "25-50" ~ "37.5",

Cover == "50-75" ~ "62.5",

Cover == ">75" ~ "87.5",

TRUE ~ "oups" (On remplace les fourchettes de couverture par un nombre )

),

Cover\_classes=as.numeric(Cover\_classes),

Cover\_max = case\_when(

Cover == "1-5" ~ "5",

Cover == "<1" ~ "1",

Cover == "5-15" ~ "15",

Cover == "15-25" ~ "25",

Cover == "25-50" ~ "50",

Cover == "50-75" ~ "75",

Cover == ">75" ~ "100",

TRUE ~ "oups" (Ici on prend le max de couverture pour chaque fourchette)

),

Cover\_max=as.numeric(Cover\_max) (On l’appelle cover\_max)

) %>%

left\_join(., Vegetation\_by\_quadrat, by = "Quadrat") %>% (on fait un left join pour coller

select(-Cover) %>% (j’élimine ceux qui n’ont pas de cover puis je groupe par quadrat)

group\_by(Quadrat) %>%

mutate(Total\_cover = sum(Cover\_classes), (Dans total\_cover je somme les couvertures de chaque plante de chaque quadrat et je pourrai donc vérifier si ça correspond bien à la couverture de de la végétation)

Total\_cover\_max = sum(Cover\_max)) %>% (Ici je somme pour chaque quadrat les couvertures maximales de chaque fourchette)

ungroup() %>%

mutate(Relative\_abundance = Cover\_classes/Total\_cover) (On définit l’abondance relative par la couverture sur la couverture totale du quadrat)

}

#Read one grid---- (Script pour lire une seule grille d’un fichier)

read\_a\_given\_sheet("Data/Echantillonnage EDGG 21\_CAM.xlsx", "EDGG3") -> df\_rel\_ab

read\_a\_given\_sheet("Data/Echantillonnage EDGG 20\_BDX.xlsx", "EDGG3") -> df\_rel\_ab

#Read all grids of a given place and year----(Script pour lire toutes les grilles d’une localité et année précises)

#List grids

readxl::excel\_sheets("Data/Echantillonnage EDGG 20\_BDX.xlsx") %>%

subset(., str\_starts(string=.,pattern="EDGG")) -> sheets (En lisant uniquement les feuilles qui commencent par EDGG)

#Apply the function to each grid in the list

lapply(sheets, function(sn) (Le script qui est sous forme de formule sera appliquée sur toutes les grilles du fichier)

read\_a\_given\_sheet("Data/Echantillonnage EDGG 20\_BDX.xlsx", sn)) %>%

do.call(rbind, .) -> df\_rel\_ab

#Read all grids of several places and years----

#Establish the list of files to be read (On display ici la liste de tous les fichiers qui contiennent xls ou Echant mais qui ne contiennent pas REU)

list.files("./", full.names = T, recursive=T) %>%

subset(., str\_detect(.,"xls")) %>%

subset(., str\_detect(.,"Echant")) %>%

subset(., str\_detect(.,"REU", negate = TRUE))-> liste\_fichiers

#Apply the function to each file in the list and each grid in each file (On applique notre function sur chaque fichier dans la liste et donc chaque grille de chaque fichier)

lapply(liste\_fichiers, function(nom\_fichier){

#List grids in the considered file

readxl::excel\_sheets(nom\_fichier) %>%

subset(., str\_starts(string=.,pattern="EDGG")) -> sheets (Dans ces fichiers je considère uniquement les feuilles qui ont EDGG)

#Apply the function to all grides in the file

lapply(sheets, function(sn)

read\_a\_given\_sheet(nom\_fichier, sn)) %>%

do.call(rbind, .)}) %>%

do.call(rbind, .) %>%

arrange(., Grid, Quadrat, -Cover\_classes) %>%

group\_by(Grid, Quadrat) %>%

mutate(

Plant\_richness = length(Plant),

Plant\_diversity = exp(-sum(Relative\_abundance\*log(Relative\_abundance))),

Rank = seq(1,length(Plant))

) %>%

ungroup() -> plant\_data

#Simplified version of the data set to focus on features of quadrats (not plants within quadrats)

plant\_data %>%

select(Year,

Locality,

Grid,

Quadrat,

Vegetation,

Total\_cover,

Total\_cover\_max,

Plant\_richness,

Plant\_diversity) %>%

distinct() -> quadrat\_data

#Graphs----

##Distribution of total cover----

ggplot(quadrat\_data, aes(fill=Locality, y=Total\_cover, x=Grid))+

geom\_boxplot(position="dodge")+

theme(axis.text.x = element\_text(angle=90))+

ggtitle("Distribution of total cover over all grids of ANG, BDX and CAM")

ggsave("Distrib\_total\_cover\_by\_grid.pdf",width = 20, height = 15, units = "cm")

ggplot(quadrat\_data, aes(x=Total\_cover, fill=Locality))+

geom\_histogram(position="dodge")+

ggtitle("Distribution of total cover over all grids of ANG, BDX and CAM")

ggsave("Distrib\_total\_cover\_by\_loc.pdf",width = 20, height = 15, units = "cm")

##Correspondence between Vegetation and Total\_cover----

ggplot(quadrat\_data, aes(x=Vegetation, y=Total\_cover\_max))+

geom\_point(position=position\_jitter(0.1))+

geom\_boxplot()+

theme(axis.text.x = element\_text(angle=90))+

ggtitle("Vegetation vs total cover")

ggsave("Vegetation\_vs\_total\_cover.pdf",width = 20, height = 15, units = "cm")

##Relationship between diversity and richness----

ggplot(quadrat\_data, aes(x=Plant\_richness, y=Plant\_diversity))+

geom\_point()+

geom\_smooth(method="lm", se=T)+

theme(axis.text.x = element\_text(angle=90))+

ggtitle("Richness-diversity")

ggsave("Richness\_diversity.pdf",width = 20, height = 15, units = "cm")

ggplot(quadrat\_data, aes(x=Plant\_richness, y=Plant\_diversity))+

geom\_point()+

geom\_smooth(method="lm", se=T)+

theme(axis.text.x = element\_text(angle=90))+

facet\_grid(~Locality)+

ggtitle("Richness-diversity")

ggsave("Richness\_diversity\_by\_loc.pdf",width = 20, height = 15, units = "cm")

##Distributions of abundances----

ggplot(plant\_data, aes(x=Cover\_classes))+

geom\_histogram()+

ggtitle("Distribution of cover classes over all grids of ANG, BDX and CAM")

ggsave("Distrib\_cover\_classes.pdf",width = 20, height = 15, units = "cm")

ggplot(plant\_data, aes(x=Cover\_classes, fill=Locality))+

geom\_histogram(position="dodge")+

ggtitle("Distribution of cover classes over all grids of ANG, BDX and CAM")

ggsave("Distrib\_cover\_by\_loc.pdf",width = 20, height = 15, units = "cm")

ggplot(plant\_data, aes(x=Relative\_abundance))+

geom\_histogram()+

ggtitle("Distribution of relative abundances over all grids of ANG, BDX and CAM")

ggsave("Distrib\_rel\_ab.pdf",width = 20, height = 15, units = "cm")

ggplot(plant\_data, aes(x=log10(Relative\_abundance)))+

geom\_histogram()+

ggtitle("Distribution of log(relative abundances) over all grids of ANG, BDX and CAM")

ggsave("Distrib\_rel\_abLOG.pdf",width = 20, height = 15, units = "cm")

ggplot(plant\_data, aes(x=Relative\_abundance, fill=Locality))+

geom\_histogram()+

ggtitle("Distribution of relative abundances over all grids of ANG, BDX and CAM")

ggsave("Distrib\_rel\_ab\_by\_loc.pdf",width = 20, height = 15, units = "cm")

##Abundance profiles----

ggplot(plant\_data, aes(x=Rank, y=Relative\_abundance))+

geom\_point()+

stat\_summary(fun.y=mean, geom="point", shape=20, size=3, color="orange", fill="orange") +

theme(axis.text.x = element\_text(angle=90))+

facet\_grid(~Locality)+

ggtitle("Abundance profile")

ggplot(plant\_data, aes(x=Rank, y=Relative\_abundance))+

geom\_point()+

stat\_summary(fun=median, geom="line", size = 1, color="orange") +

theme(axis.text.x = element\_text(angle=90))+

ggtitle("Abundance profiles")

ggsave("Abundance\_profile.pdf",width = 20, height = 15, units = "cm")

ggplot(plant\_data, aes(x=Rank, y=Relative\_abundance, color=Plant\_richness))+

geom\_point()+

theme(axis.text.x = element\_text(angle=90))+

facet\_wrap(~Plant\_richness)+

ggtitle("Abundance profiles in each richness class")

ggsave("Abundance\_profile\_by\_richness.pdf",width = 20, height = 15, units = "cm")

ggplot(plant\_data, aes(x=Rank, y=Relative\_abundance, col = Locality))+

geom\_point()+

stat\_summary(fun=median, geom="line", size=1, color="black") +

theme(axis.text.x = element\_text(angle=90))+

facet\_grid(~Locality)+

ggtitle("Abundance profiles")

ggsave("Abundance\_profile\_by\_loc.pdf",width = 20, height = 15, units = "cm")

ggplot(plant\_data, aes(x=Rank, y=Cover\_classes))+

geom\_point(position = position\_jitter(width = 0.25))+

theme(axis.text.x = element\_text(angle=90))+

ggtitle("Abundance profile")