

Extraction des données SAMM et mise en forme pour un modèle d'occupancy

Valentin Lauret

09/10/2020

Mise en forme des données SAMM pour l'étude d'occupancy multi-espèces grand dauphins - chalutiers.

Les données

J'utilise les données SAMM que Matthieu a récemment envoyé pour les observations de grand dauphin et des activités de pêche. Je prends les données d'effort utilisées pour l'étude Occupancy.

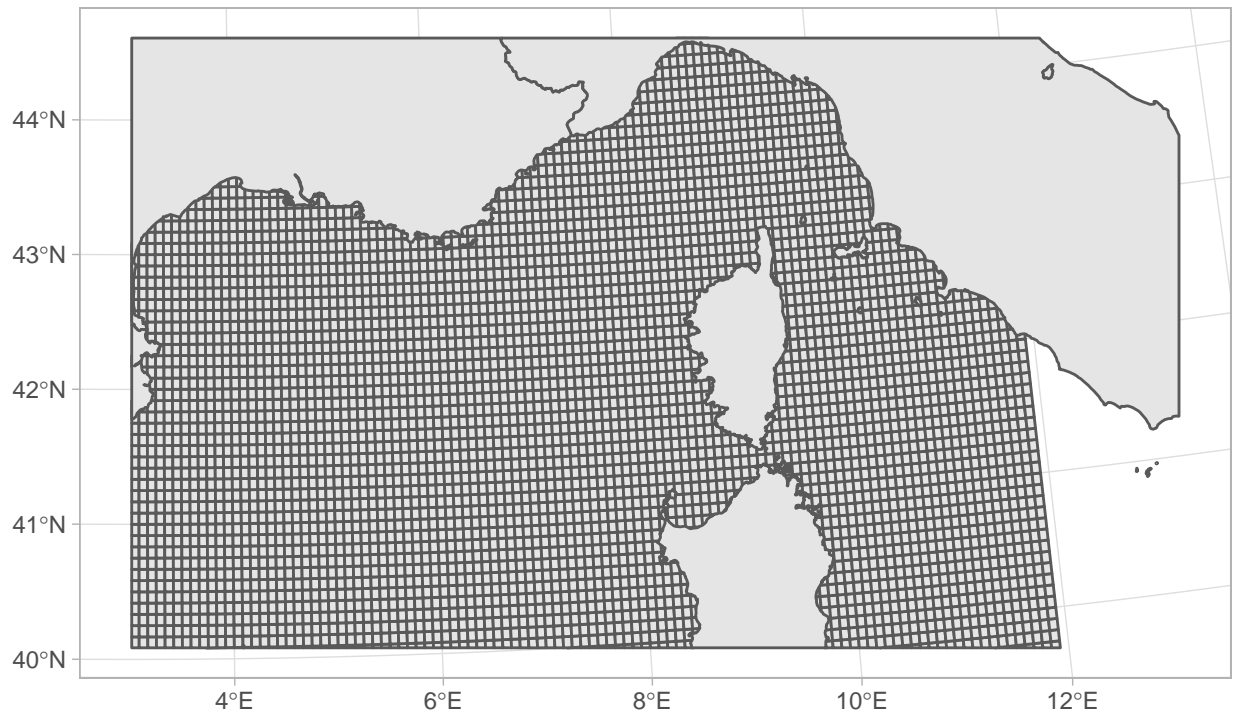
```
load("20180914_SAMM_data-LauretValentin.RData") # données dauphins Matthieu
load("trSAMM.rdata") # donnees d'effort utilisées pour l'occupancy
```

La grille d'étude

La grille.

```
load("pays.rdata")
grid <- st_read("Grid/grid.shp") %>% st_transform(crs = st_crs(pays))

## Reading layer `grid' from data source `/Users/valentinlauret/Google Drive/These/Work/Multispecies Oc
## Simple feature collection with 4356 features and 3 fields
## geometry type: POLYGON
## dimension: XY
## bbox: xmin: 701000 ymin: 5886622 xmax: 1467639 ymax: 6390000
## epsg (SRID): NA
## proj4string: +proj=lcc +lat_1=44 +lat_2=49 +lat_0=46.5 +lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80
grid %>%
  ggplot() +
  geom_sf() +
  geom_sf(data= pays)
```

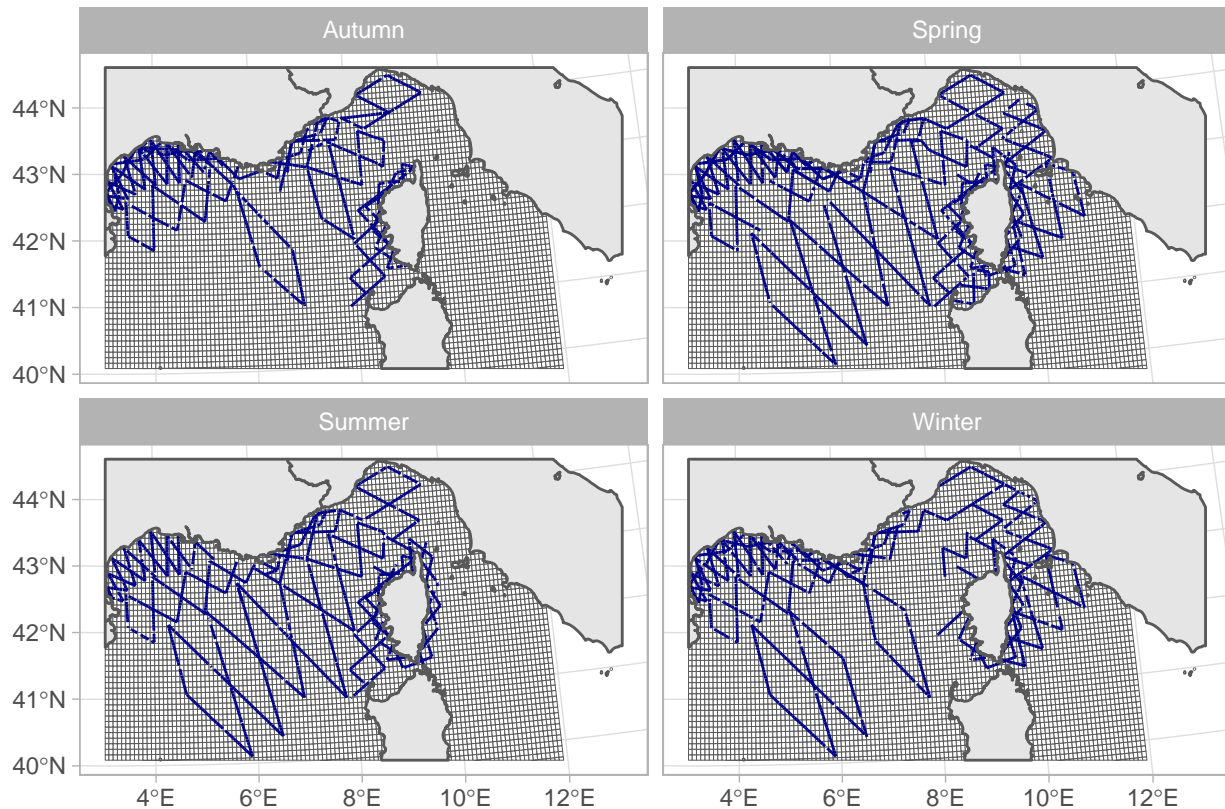


Les transects

Visualisation des transects par saison.

```
tr2 <- tr %>% mutate(season = case_when(
  m %in% c(1,2,3) ~ "Winter",
  m %in% c(4,5,6) ~ "Spring",
  m %in% c(7,8,9) ~ "Summer",
  m %in% c(10,11,12) ~ "Autumn"
))

grid %>%
  ggplot() +
  geom_sf(fill = "white", lwd = 0.1) +
  geom_sf(data= tr2, color = "darkblue")+
  geom_sf(data= pays) +
  facet_wrap(~season, ncol = 2)
```



Les observations de grand dauphin

J'utilise la table `$segdata` en filtrant les entrées qui ont `count > 0` pour garder uniquement les lignes avec observation de dauphins.

```
seg <- bind_rows(winter$segdata, summer$segdata) %>% mutate(season = c(rep("winter", nrow(winter$segdata),
obs <- seg %>% filter(count > 0))
```

Nombre de dauphins observés

```
obs %>% select(count) %>% sum() # 498 dauphins
```

```
## [1] 498
```

Nombre d'observations de dauphins

```
obs %>% nrow() # 78 observations
```

```
## [1] 78
```

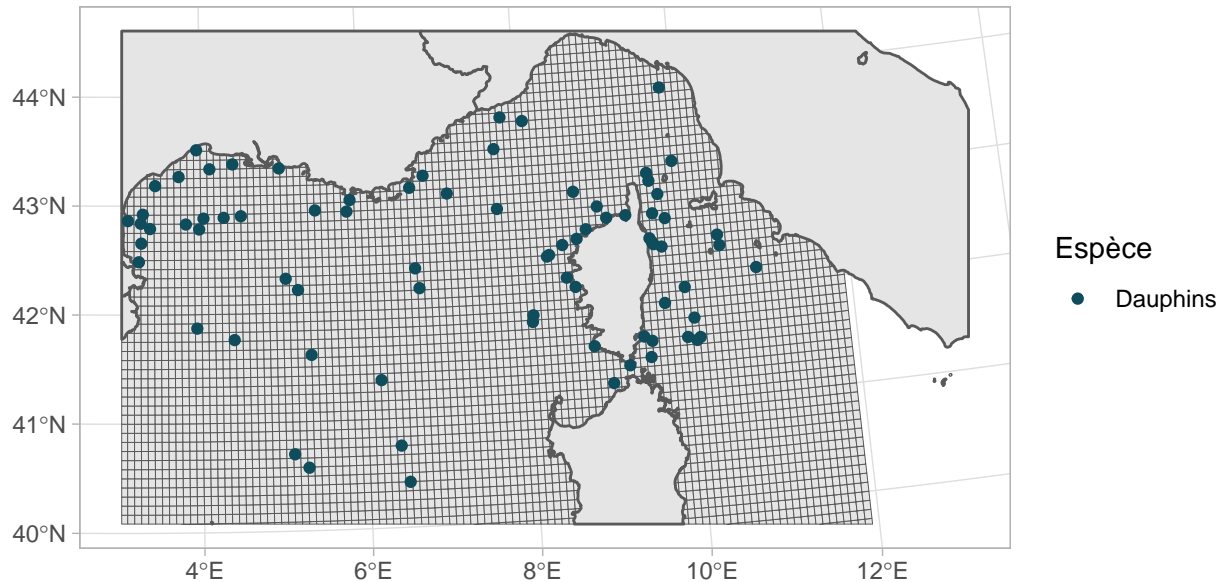
Visualisation spatiale ds detections de dauphin.

```
sobs <- obs %>% st_as_sf(coords = c("X", "Y"), crs = st_crs(grid))

grid %>% ggplot() + geom_sf(lwd = 0.1)+
  geom_sf(data = pays) +
  geom_sf(data = sobs, aes(color = "Dauphins")) +
  scale_color_manual(name = "Espèce", values = c("Dauphins" = "#0F4C5C")) +
  labs(title = "Observations de grand dauphin", subtitle = "Données SAMM été et hiver")
```

Observations de grand dauphin

Données SAMM été et hiver



Données de pêche

Charge les données sur les activités humaines

```
load("20200928_SAMM_data_Pressure.RData")
```

```
head(winter$obsdata)
```

##	TRANSECT	Transect.Label	Region.Label	Sample.Label	distance	size	object
## 12874	C2/06	522	MED_C	587	0.03889785	1	13059
## 12875	C2/06	522	MED_C	587	0.08533430	1	13060
## 12877	C2/06	522	MED_C	587	0.03889785	1	13062
## 12878	C2/06	522	MED_C	587	0.08533430	1	13063
## 12882	C2/06	522	MED_C	587	0.04903470	1	13067
## 13235	C2/09	547	MED_C	619	0.22598618	1	13420

Problème : il manque les dates de chaque observation. On va faire le lien via les `Sample.Label` de la table `seg` qui contient les transects.

```
# winter
activ_w <- winter_fishingactivities$obsdata %>%
  mutate(Sample.Label = as.numeric(Sample.Label)) %>%
  mutate(date = as_date(NA))

seg_dat <- seg %>% filter(season == "winter") %>% select(date, Sample.Label) %>%
  mutate(Sample.Label = as.numeric(Sample.Label))
```

```

for(i in 1:nrow(activ_w)){

  index <- which(seg_dat$Sample.Label == activ_w$Sample.Label[i])[1]

  activ_w$date[i] <- as_date(seg_dat$date[index])
}

# summer
activ_s <- summer_fishingactivities$obsdata %>%
  mutate(Sample.Label = as.numeric(Sample.Label)) %>%
  mutate(date = as_date(NA))

seg_dat <- seg %>% filter(season == "summer") %>% select(date, Sample.Label) %>%
  mutate(Sample.Label = as.numeric(Sample.Label))

for(i in 1:nrow(activ_s)){

  index <- which(seg_dat$Sample.Label == activ_s$Sample.Label[i])[1]

  activ_s$date[i] <- seg_dat$date[index]
}

```

Maintenant qu'on a rajouté les dates on peut fusionner les tables. NB. Il fallait rajouter les dates AVANT de fusionner les tables car il y a des `Sample.Label` identiques en été et en hiver. En gros, ils ont repris à 0 la numérotation des `Sample.Label` entre l'été et l'hiver.

```

activ <- bind_rows(activ_w, activ_s) %>%
  mutate(season = c(rep("winter", nrow(activ_w)), rep("summer", nrow(activ_s))))

```

Le nombre d'évènement par type d'activité de pêche

```

activ %>%
  count(what)

```

```

##              what      n
## 1 Bateau art dormant (fileyeur, caseyeur) 19
## 2              Bateau chalutier 122
## 3              Bateau de peche pro 11
## 4          Bateau senneur, bolincheur 2
## 5              Bouee de peche 299

```

Pour l'étude, on ne garde que les chalutiers. On filtre sur `what == "Bateau chalutier"` et on spatialise avec les colonnes `latitude` et `longitude`.

```

sactiv <- activ %>%
  filter(what == "Bateau chalutier") %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = "+proj=longlat +datum=WGS84") %>%
  st_transform(crs = st_crs(grid))

grid %>% ggplot() + geom_sf(lwd = 0.1, fill = "white") +
  geom_sf(data = pays) +
  geom_sf(data = tr2, aes(color = "Transects"), alpha = 0.4) +
  geom_sf(data = sobs, aes(color = "Dauphins"), alpha = 0.7) +
  geom_sf(data = sactiv, aes(color = "Chalutiers"), alpha = 0.7) +

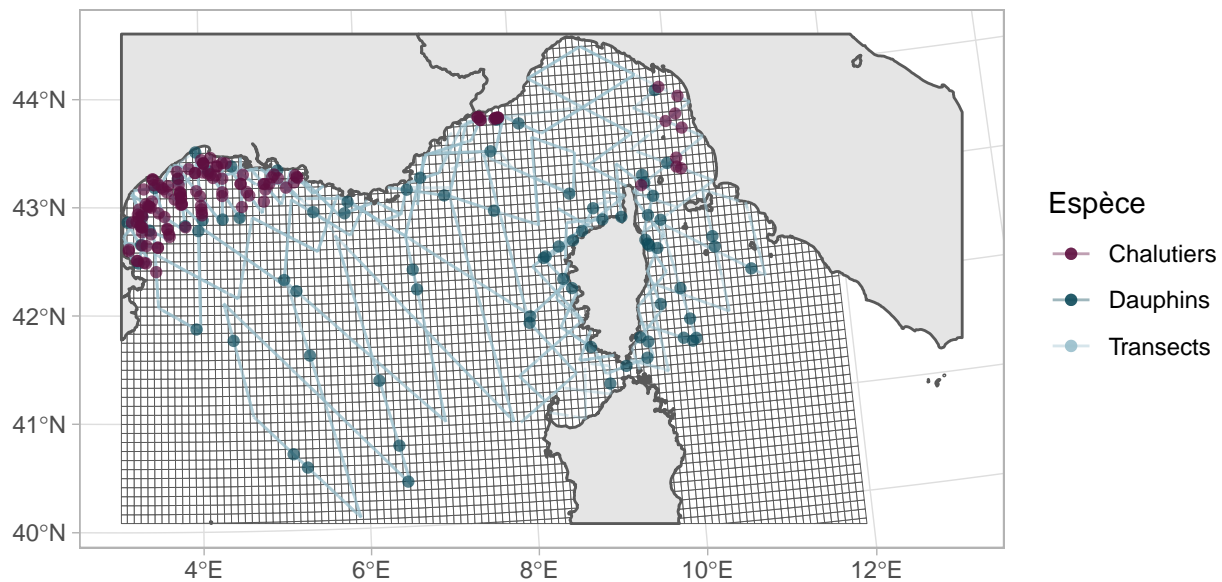
```

```
scale_color_manual(name = "Espèce", values = c("Dauphins" = "#0F4C5C", "Chalutiers" = "#5f0f40", "Transects" = "#5f0f40"),
labs(title = "Observations de chalutiers", subtitle = "Données SAMM été et hiver \nObs de dauphin en arrièr
```

Observations de chalutiers

Données SAMM été et hiver

Obs de dauphin en arrière plan



Création des matrices pour l'occupancy

On va créer 3 matrices de dimensions nb_cellules x nb_occasions, i.e. 4356 x 4 :

- `samm_effort` pour stocker l'effort comme la longueur de transect par cellule et par occasion.
- `yd_samm` pour stocker les détections de **grand dauphin** par cellule et par occasion.
- `yp_samm` pour stocker les détections de **chalutier** par cellule et par occasion.

Observation de grand dauphin

Intersection des observations de grand dauphin avec les cellules de la grille occupancy

```
int <- sobs %>%
  st_intersection(grid) %>%
  select(date, Sample.Label, count, season, objectid, geometry)

int %>% st_intersects(grid) %>% # nombre de cellule de la grille avec une detection
  unlist() %>%
  unique() %>%
  length()
```

```
## [1] 77
```

Sépare chaque observation en fonction du mois de l'année pour l'assigner à une occasion.

```
int2 <- int %>%
  mutate(month = month(date)) %>%
  mutate(month = as_factor(month),
         month = fct_relevel(month, c("11", "12", "1", "2", "5", "6", "7", "8")))

# cree la matrice
yd_samm <- grid %>% mutate(obs = 0, autumn = 0, winter = 0, spring = 0, summer = 0)

# remplit la matrice (il y a peut-être un version tidy pour reste ça rapidos)
for(i in 1:nrow(int2)){

  index <- which(yd_samm$objectid == int2$objectid[i])
  occ <- int2$month[i]

  yd_samm$obs[index] <- 1

  if( occ %in% c("1","2","3") ) yd_samm$winter[index] <- 1
  if( occ %in% c("4","5","6") ) yd_samm$spring[index] <- 1
  if( occ %in% c("7","8","9") ) yd_samm$summer[index] <- 1
  if( occ %in% c("10","11","12") ) yd_samm$autumn[index] <- 1

}

# verif
sum(yd_samm$obs)
```

```
## [1] 77
```

La matrice observation de grand dauphin.

```
head(yd_samm)
```

```
## Simple feature collection with 6 features and 8 fields
## geometry type: POLYGON
## dimension: XY
## bbox: xmin: 701000 ymin: 5886622 xmax: 707143.7 ymax: 5942119
## epsg (SRID): NA
## proj4string: +proj=lcc +lat_1=44 +lat_2=49 +lat_0=46.5 +lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80
##   objectid ligne colonne geometry obs autumn winter
## 1 22240 EY 133 POLYGON ((701000 5895635, 7... 0 0 0
## 2 22241 EX 133 POLYGON ((701000 5904934, 7... 0 0 0
## 3 22242 EW 133 POLYGON ((701000 5914231, 7... 0 0 0
## 4 22243 EV 133 POLYGON ((701000 5923527, 7... 0 0 0
## 5 22244 EU 133 POLYGON ((701000 5932822, 7... 0 0 0
## 6 22245 ET 133 POLYGON ((701000 5942116, 7... 0 0 0
##   spring summer
## 1 0 0
## 2 0 0
## 3 0 0
## 4 0 0
## 5 0 0
## 6 0 0
```


Observation de chalutiers

Intersection des observations de chalutiers avec les cellules de la grille occupancy

```
int <- sactiv %>%
  st_intersection(grid) %>%
  select(date, Sample.Label, what, season, objectid, geometry)

int %>% st_intersects(grid) %>% # nombre de cellule de la grille avec une detection
  unlist() %>%
  unique() %>%
  length()
```

```
## [1] 72
```

Sépare chaque observation en fonction du mois de l'année pour l'assigner à une occasion.

```
int2 <- int %>%
  mutate(month = month(date)) %>%
  mutate(month = as_factor(month))

# cree la matrice
yp_samm <- grid %>% mutate(obs = 0, autumn = 0, winter = 0, spring = 0, summer = 0)

# remplit la matrice (il y a peut-être un version tidy pour reste ça rapidos)
for(i in 1:nrow(int2)){

  index <- which(yp_samm$objectid == int2$objectid[i])
  occ <- int2$month[i]

  yp_samm$obs[index] <- 1

  if( occ %in% c("1","2","3") ) yp_samm$winter[index] <- 1
  if( occ %in% c("4","5","6") ) yp_samm$spring[index] <- 1
  if( occ %in% c("7","8","9") ) yp_samm$summer[index] <- 1
  if( occ %in% c("10","11","12") ) yp_samm$autumn[index] <- 1

}

# verif
sum(yp_samm$obs)
```

```
## [1] 72
```

La matrice observation de chalutier.

```
head(yp_samm)
```

```
## Simple feature collection with 6 features and 8 fields
## geometry type: POLYGON
## dimension: XY
## bbox: xmin: 701000 ymin: 5886622 xmax: 707143.7 ymax: 5942119
## epsg (SRID): NA
## proj4string: +proj=lcc +lat_1=44 +lat_2=49 +lat_0=46.5 +lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80
## objectid ligne colonne geometry obs autumn winter
## 1 22240 EY 133 POLYGON ((701000 5895635, 7... 0 0 0
## 2 22241 EX 133 POLYGON ((701000 5904934, 7... 0 0 0
## 3 22242 EW 133 POLYGON ((701000 5914231, 7... 0 0 0
```



```
## 4      22243      EV      133 POLYGON ((701000 5923527, 7... 0      0      0
## 5      22244      EU      133 POLYGON ((701000 5932822, 7... 0      0      0
## 6      22245      ET      133 POLYGON ((701000 5942116, 7... 0      0      0
##      spring summer
## 1          0      0
## 2          0      0
## 3          0      0
## 4          0      0
## 5          0      0
## 6          0      0
```

Effort d'échantillonnage

Longueur totale des transect SAMM été et hiver

```
st_length(tr2) %>% sum()
```

```
## 29346866 [m]
```

Intersection entre tous les transects et les cellules de la grille + calcul de la longueur de chaque bout de transect

```
int <- tr2 %>%
  st_intersection(grid) # %>%
```

```
int <- int %>% select(m, season, objectid, geometry) %>% mutate(eff = st_length(int))
```

Dans plusieurs cas, il y a différents transects qui sont passés dans une même cellule au cours de la même saison. Il faut regrouper.

```
int2 <- int %>%
  group_by(objectid, season) %>%
  summarise(sum(eff))
```

```
# nb de sites différents échantillonnés
length(unique(int2$objectid))
```

```
## [1] 1335
```

```
# effort total
sum(int2$`sum(eff)`)
```

```
## 29346866 [m]
```

Maintenant on peut séparer chaque transect en fonction du mois de l'année pour l'assigner à une occasion.

```
# cree la matrice
```

```
effort_samm <- grid %>% mutate(eff.tot = 0, autumn = 0, winter = 0, spring = 0, summer = 0)
```

```
# remplit la matrice (il y a peut-être un version tidy pour reste ça rapides)
for(i in 1:nrow(int2)){
```

```
  index <- which(effort_samm$objectid == int2$objectid[i])
  occ <- int2$season[i]
```

```
  if( occ == "Winter" )   effort_samm$winter[index] <- int2$`sum(eff)`[i]
  if( occ == "Spring" )  effort_samm$spring[index] <- int2$`sum(eff)`[i]
  if( occ == "Summer" )  effort_samm$summer[index] <- int2$`sum(eff)`[i]
  if( occ == "Autumn" )  effort_samm$autumn[index] <- int2$`sum(eff)`[i]
```

```

}

effort_samm$eff.tot <- effort_samm$autumn + effort_samm$winter + effort_samm$summer + effort_samm$spring

# verif
sum(effort_samm$eff.tot)

## [1] 29346866

La matrice de l'effort d'échantillonnage.

head(effort_samm)

## Simple feature collection with 6 features and 8 fields
## geometry type: POLYGON
## dimension: XY
## bbox: xmin: 701000 ymin: 5886622 xmax: 707143.7 ymax: 5942119
## epsg (SRID): NA
## proj4string: +proj=lcc +lat_1=44 +lat_2=49 +lat_0=46.5 +lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80
##   objectid ligne colonne geometry eff.tot autumn winter
## 1 22240 EY 133 POLYGON ((701000 5895635, 707143.7 5895635, 707143.7 5942119, 701000 5942119, 701000 5895635)) 0 0 0
## 2 22241 EX 133 POLYGON ((701000 5904934, 707143.7 5904934, 707143.7 5942119, 701000 5942119, 701000 5904934)) 0 0 0
## 3 22242 EW 133 POLYGON ((701000 5914231, 707143.7 5914231, 707143.7 5942119, 701000 5942119, 701000 5914231)) 0 0 0
## 4 22243 EV 133 POLYGON ((701000 5923527, 707143.7 5923527, 707143.7 5942119, 701000 5942119, 701000 5923527)) 0 0 0
## 5 22244 EU 133 POLYGON ((701000 5932822, 707143.7 5932822, 707143.7 5942119, 701000 5942119, 701000 5932822)) 0 0 0
## 6 22245 ET 133 POLYGON ((701000 5942116, 707143.7 5942116, 707143.7 5942119, 701000 5942119, 701000 5942116)) 0 0 0
##   spring summer
## 1 0 0
## 2 0 0
## 3 0 0
## 4 0 0
## 5 0 0
## 6 0 0

```

Résumé

Regroupe les trois matrices dans une liste et sauve les résultats

```

multioccu_samm <- list(dauphins = yd_samm, chalut = yp_samm, effort = effort_samm)

save(multioccu_samm, file = "msoccu_samm.rdata")

```

Visualisation des occurrences des deux espèces

```

# co-occurrence
cooc <- which(multioccu_samm$chalut$obs == 1 & multioccu_samm$dauphins$obs == 1)

# plot
multioccu_samm$dauphins[] %>% filter(obs == 1) %>% ggplot() + geom_sf(aes(fill = "Dauphins"), lwd = 0.1) +
  geom_sf(data = multioccu_samm$chalut[multioccu_samm$chalut$obs == 1,], aes(fill = "Chalutiers"), lwd = 0.1) +
  geom_sf(data = grid[cooc,], aes(fill = "Co-occurrence"), lwd = 0.1) +
  scale_fill_manual(name = "Espèce", values = c("Dauphins" = "#0F4C5C", "Chalutiers" = "#5f0f40", "Co-occurrence" = "#f08080")) +
  geom_sf(data = pays) +
  labs(title = "Occurrences des dauphins et chalutiers à partir des données SAMM",
        subtitle = "sur la grille utilisée pour l'occupancy")

```

Occurrences des dauphins et chalutiers à partir des données SAMM
sur la grille utilisée pour l'occupancy

