



Ecole nationale de la Statsitique et de l'Analyse économique

Année académique 2023-2024

* * *

COURS DE STATISTIQUE SPATIALE

* * *

TRAVAUX PRATIQUES N°2

* * *

Réalisé par : **Inoussa ZAONGO et Omar THIAM**

Sous la supervision de : **M. HEMA**

21 Décembre 2023

Ce document a été généré directement depuis **RStudio** en utilisant l'outil **Markdown**.



Sommaire

Importation et visualisation des données ACLED	3
Ajouter des coordonnées spatiales à la base	4
Ajout de la projection	4
Transformation de l'objet sp en sf	4
Representation	5
Importation des shapefile des pays de l'Afrique	6
Selectionner les limites des pays de ACLED dans le sf de l'Afrique	6
Representation des pays et des evenements de ACLED	6
I.Première partie	7
1. Compter les événements dans chaque pays	7
2.Fixation d'un pays et Calcul des événements	8
2.1 Evénements par region	8
Représentation	9
Extraction de la base Burkina dans la base Acled	10
Fusion de base BF et de base BF_ACLED	10
2.2 Evénements par province	13
Fusion de base BF_provin et de base BF_ACLED	14
II Deuxième partie	15
II.1.Créer un raster avec une résolution de 0.5m x 0.5 m	15
II.2.creer un autre raster qui categorise le nombre d'événements	17
II.2.1 Categorisation du prof	17
II.2.1 Categorisation differente de celle du prof	20
II.3 BONUS	23
II.3.1 Resolution 10km*10km	23
II.3.2 Carte du G5_sahel	24
II.3.3 Matrice de voisinage	25
II.3.3.1 Sur les provinces du Burkina Faso	25
II.3.3.1 Sur les departements du Senegal	28
Representation graphique de la matrice de voisinage du senegal	28

```
library("raster")
library("sf")
library("dplyr")
library("ggplot2")
library("sp")
#library("esquisse")
library("readr")
```

Importation et visualisation des données ACLED

Ajouter des coordonnées spatiales à la base

```
data_sp = data
coordinates(data_sp) = ~longitude+latitude
class(data_sp) # pour connaître la classe

## [1] "SpatialPointsDataFrame"
## attr(,"package")
## [1] "sp"
#View(data_sp)
```

Ajout de la projection

```
# projection longlat
proj4string(data_sp) <- CRS("+proj=longlat +datum=WGS84 +no_defs
+ellps=WGS84 +towgs84=0,0,0")
```

```
#Pour connaître le numéro de projection wgs_84
str(data_sp)
```

```
## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
##   ..@ data      : tibble [54,138 x 5] (S3: tbl_df/tbl/data.frame)
##   ... .$.id     : chr [1:54138] "BF010160" "BF010164" "BF010171" "MAA1633" ...
##   ... .$.date   : chr [1:54138] "30-Jun-23" "30-Jun-23" "30-Jun-23" "30-Jun-23" ...
##   ... .$.annee  : num [1:54138] 2023 2023 2023 2023 2023 ...
##   ... .$.type   : chr [1:54138] "Strategic developments" "Strategic developments" "Explosions/Remote v
##   ... .$.pays   : chr [1:54138] "Burkina Faso" "Burkina Faso" "Burkina Faso" "Mauritania" ...
##   ..@ coords.nrs : int [1:2] 7 6
##   ..@ coords    : num [1:54138, 1:2] -0.03 1.772 0.534 -17.038 -11.57 ...
##   ... .- attr(*, "dimnames")=List of 2
##   ...   .$.      : chr [1:54138] "1" "2" "3" "4" ...
##   ...   .$.      : chr [1:2] "longitude" "latitude"
##   ..@ bbox      : num [1:2, 1:2] -25.16 2.86 14.65 25.99
##   ... .- attr(*, "dimnames")=List of 2
##   ...   .$.      : chr [1:2] "longitude" "latitude"
##   ...   .$.      : chr [1:2] "min" "max"
##   ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
##   ... .- projargs: chr "+proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs"
##   ...   .$.comment: chr "BOUNDCRS[\n      SOURCECRS[\n          GEOGCRS[\\"unknown\\",\n
proj4string(data_sp)
```

```
## [1] "+proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs"
```

Transformation de l'objet sp en sf

```
data_sf = st_as_sf(data_sp, coords=c(longitude,latitude))
class(data_sf)
```

```
## [1] "sf"          "tbl_df"       "tbl"          "data.frame"
head(data_sf)
```

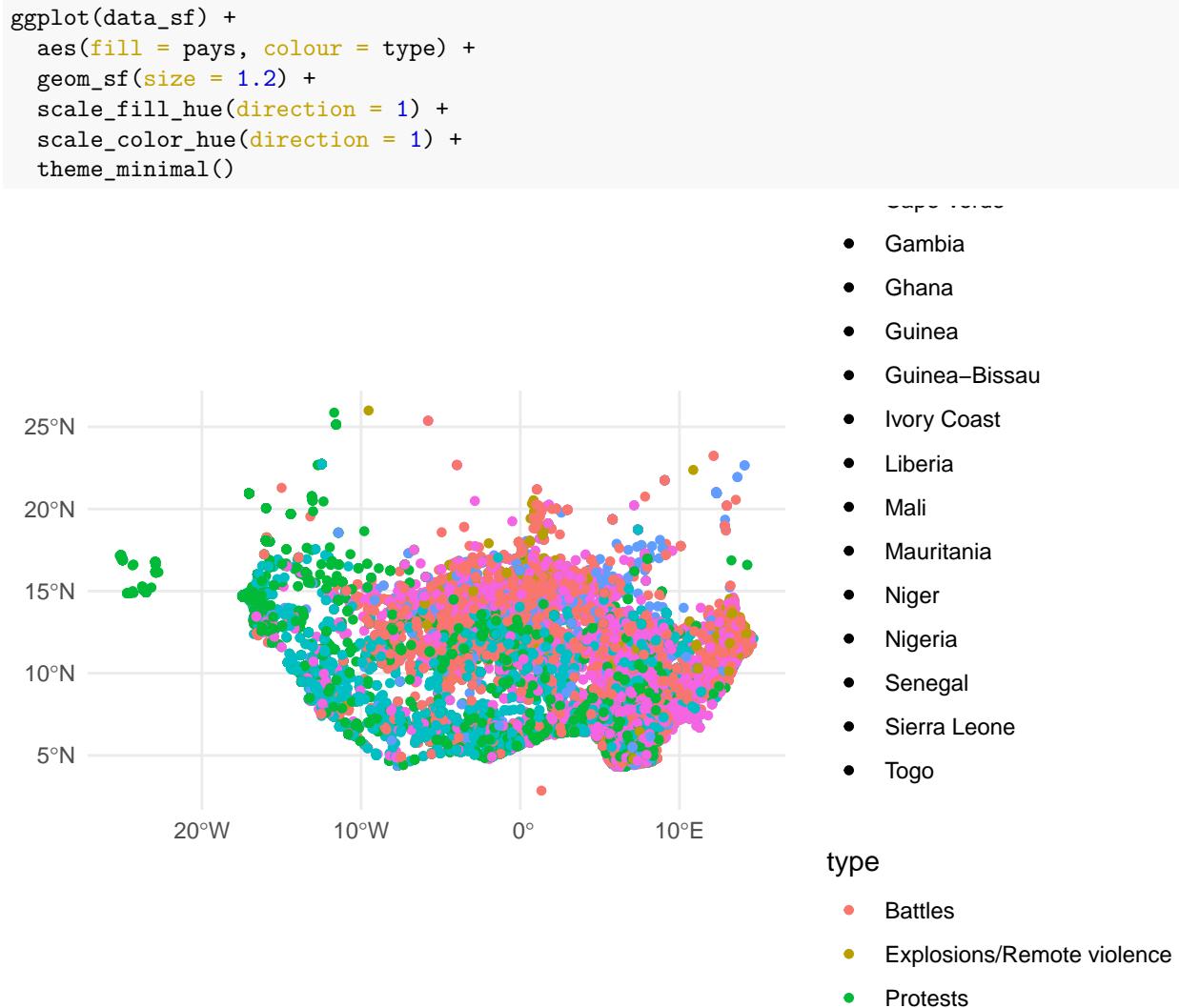
```
## Simple feature collection with 6 features and 5 fields
```

```

## Geometry type: POINT
## Dimension:      XY
## Bounding box:  xmin: -17.038 ymin: 10.9966 xmax: 1.7719 ymax: 25.14
## Geodetic CRS: +proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs
## # A tibble: 6 x 6
##   id      date    annee type          pays      geometry
##   <chr>   <chr>   <dbl> <chr>        <chr>      <POINT [°]>
## 1 BF010160 30-Jun-23 2023 Strategic developmen~ Burk~  (-0.03 14.04)
## 2 BF010164 30-Jun-23 2023 Strategic developmen~ Burk~  (1.7719 12.0299)
## 3 BF010171 30-Jun-23 2023 Explosions/Remote vi~ Burk~  (0.5344 10.9966)
## 4 MAA1633  30-Jun-23 2023 Protests          Maur~  (-17.038 20.9434)
## 5 MAA1636  30-Jun-23 2023 Protests          Maur~  (-11.57 25.14)
## 6 MLI30435 30-Jun-23 2023 Violence against civ~ Mali   (-4.1698 14.1005)

```

Representation



Importation des shapefile des pays de l'Afrique

```
pass_true <- paste0(pass,"/","Africa_Boundaries.shp") #Accés a notre base de données
Afrique<- read_sf(pass_true) #Lecture de notre base de données
```

Selectionner les limites des pays de ACLED dans le sf de l'Afrique

```
library(dplyr)

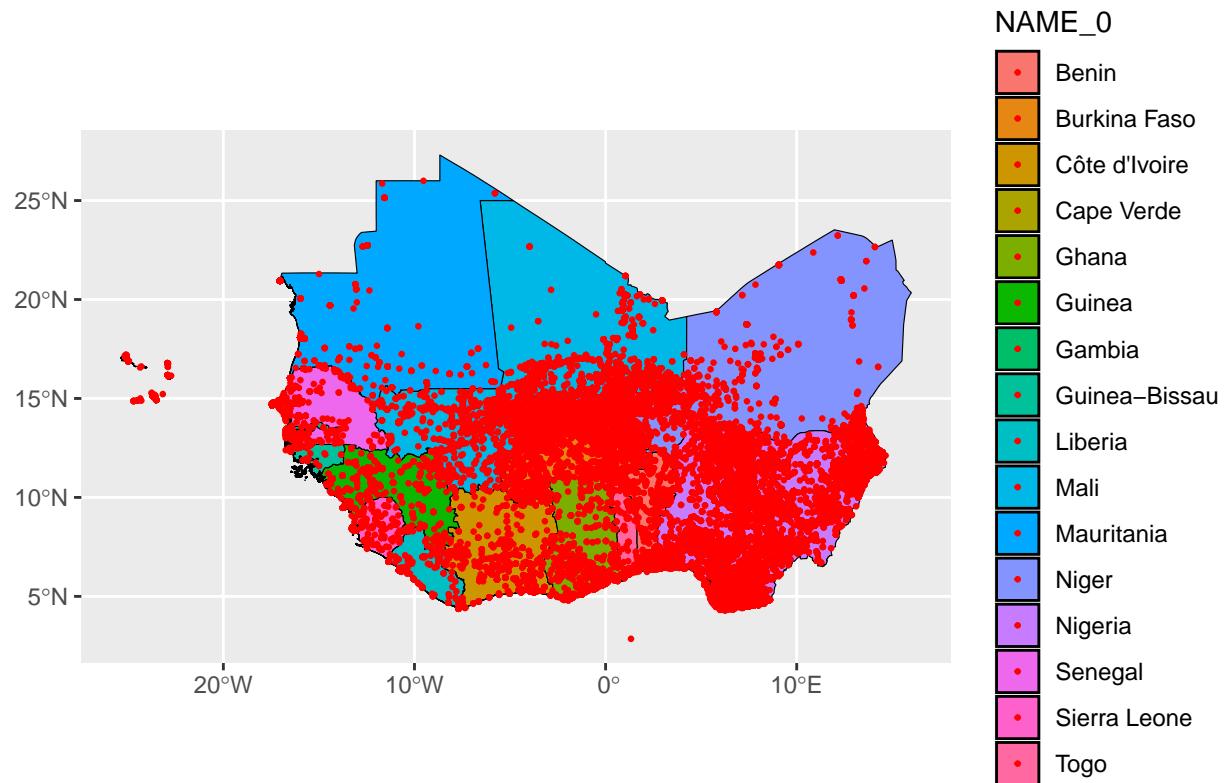
Pays_ACLED = filter(Afrique,NAME_0 %in% c("Burkina Faso","Côte d'Ivoire",
                                         "Gambia","Ghana","Liberia","Niger",
                                         "Sierra Leone","Togo","Mali","Nigeria",
                                         "Mauritania","Senegal","Benin","Cape Verde" , "Guinea","Guinea
```

Representation des pays et des evenements de ACLED

```
library(ggplot2)

graph <- ggplot() +
  geom_sf(data = Pays_ACLED, aes(fill = NAME_0), color = "black", size = 0.5) +
  geom_sf(data = data_sf, aes(fill = type), color = "red", size = 0.5) +
  scale_fill_discrete(name = "NAME_0", limits = unique(Pays_ACLED$NAME_0))

print(graph)
```



I.Première partie

1. Compter les événements dans chaque pays

#1.1 Calculer le nombre d'evenement par pays

```
library(dplyr)
nombre_evenement = data %>%
  group_by(pays) %>%
  summarise("nombre_evenement" = n())

print(nombre_evenement)

## # A tibble: 16 x 2
##   pays           nombre_evenement
##   <chr>              <int>
## 1 Benin                917
## 2 Burkina Faso         8466
## 3 Cape Verde             188
## 4 Gambia                180
## 5 Ghana                1510
## 6 Guinea                1813
## 7 Guinéa-Bissau          127
## 8 Ivory Coast            1366
## 9 Liberia                561
```

```

## 10 Mali           8018
## 11 Mauritania   1276
## 12 Niger          3110
## 13 Nigeria        24717
## 14 Senegal         1027
## 15 Sierra Leone   522
## 16 Togo            340

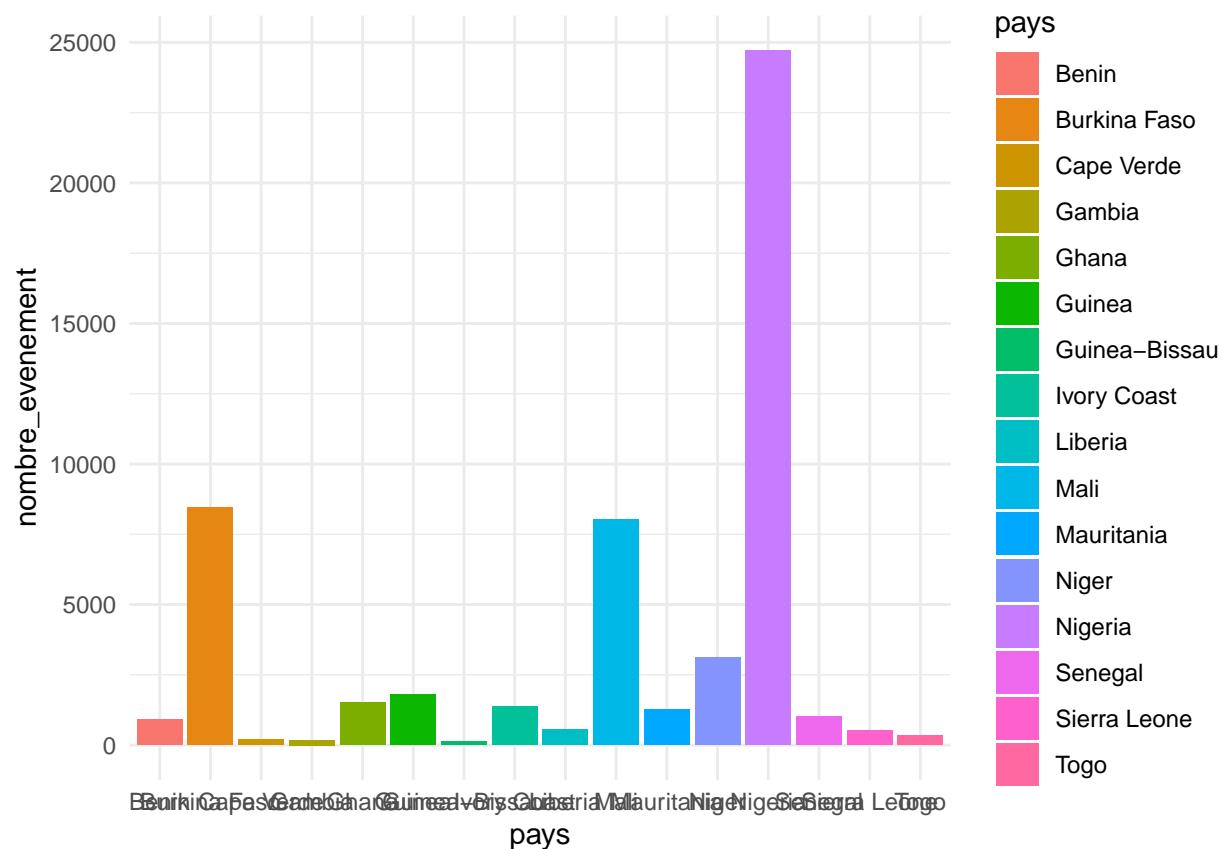
```

#1.2 Representation

```

ggplot(nombre_evenement) +
  aes(x = pays, y = nombre_evenement, fill = pays) +
  geom_col() +
  scale_fill_hue(direction = 1) +
  theme_minimal()

```



2. Fixation d'un pays et Calcul des événements

2.1 Evénements par region

Fixons comme pays le Burkina Faso et calcul les événements par region

Pour cela, nous allons faire venir le shape par region du Burkina Faso

```

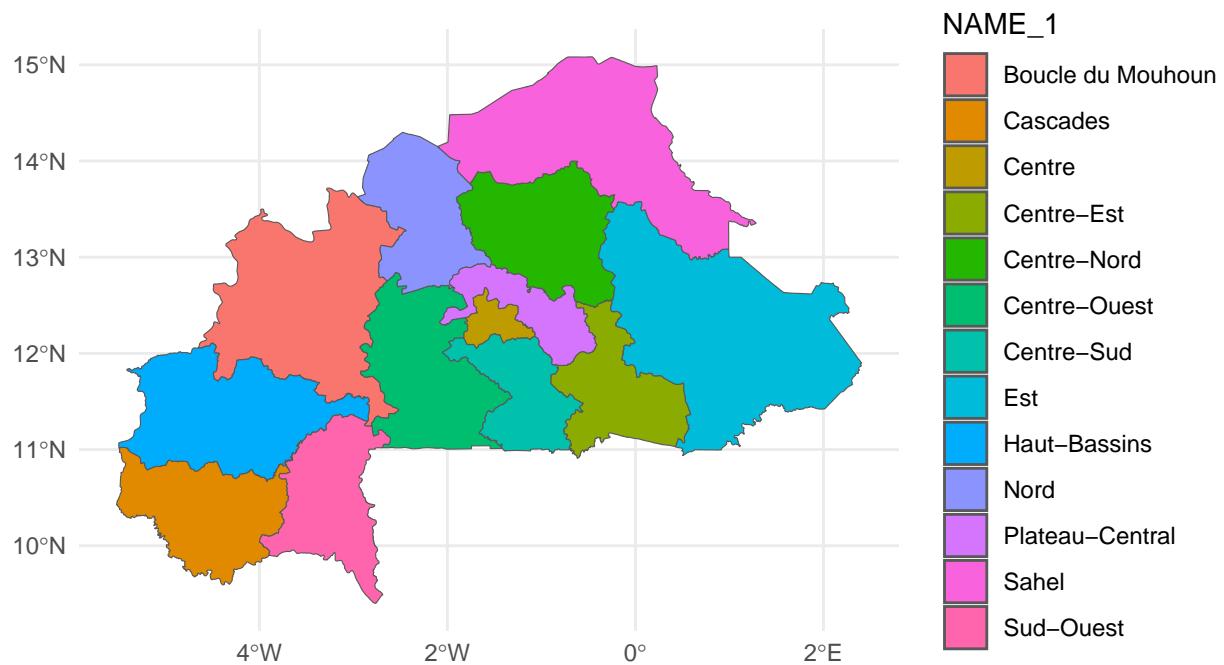
pass <- here::here() #chemin relatif
pass_true <- paste0(pass,"/","gadm41_BFA_1.shp") #Accès à notre base de données
BF<- read_sf(pass_true) #Lecture de notre base de données

```

Représentation

#representation

```
ggplot(BF) +  
  aes(fill = NAME_1) +  
  geom_sf(size = 1.2) +  
  scale_fill_hue(direction = 1) +  
  theme_minimal()
```



Extraction de la base Burkina dans la base Acled

```
# extraction de la base Burkina dans la base Acled
BF_ACLED = filter(data_sf,pays %in% ("Burkina Faso"))
```

Nous allons ensuite fusionner cette base qui contient les attaques avec la base BF qui contient le découpage administratif pour avoir le nombre d'événement.

Fusion de base BF et de base BF_ACLED

```
New_data= st_join(BF_ACLED,BF)
head(New_data)

## Simple feature collection with 6 features and 16 fields
## Geometry type: POINT
## Dimension: XY
## Bounding box: xmin: -1.6974 ymin: 10.9966 xmax: 1.7719 ymax: 14.3806
## Geodetic CRS: +proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs
## # A tibble: 6 x 17
##   id      date annee type      pays      geometry GID_1      GID_0 COUNTRY
##   <chr>   <chr> <dbl> <chr> <chr>   <POINT [°]> <chr>   <chr> <chr>
## 1 BF010160 30-J- 2023 Stra- Burk- (-0.03 14.04) BFA.12_1 BFA Burkina Faso
## 2 BF010164 30-J- 2023 Stra- Burk- (1.7719 12.0299) BFA.8_1 BFA Burkina Faso
## 3 BF010171 30-J- 2023 Expl- Burk- (0.5344 10.9966) BFA.8_1 BFA Burkina Faso
```

```

## 4 BF010121 29-J- 2023 Expl~ Burk~ (-1.5461 13.6781) BFA.4_1 BFA Burkina Faso
## 5 BF010125 29-J- 2023 Expl~ Burk~ (-1.4348 13.9068) BFA.12_1 BFA Burkina Faso
## 6 BF010126 29-J- 2023 Expl~ Burk~ (-1.6974 14.3806) BFA.12_1 BFA Burkina Faso
## # i 8 more variables: NAME_1 <chr>, VARNAME_1 <chr>, NL_NAME_1 <chr>,
## # TYPE_1 <chr>, ENGTTYPE_1 <chr>, CC_1 <chr>, HASC_1 <chr>, ISO_1 <chr>
#Calcul des événements par region au burkina faso

#AVec summarise

evenement_BF = New_data %>%
  group_by(NAME_1)%>%
  summarise("evenement_region"=n())
print(evenement_BF)

## Simple feature collection with 14 features and 2 fields
## Geometry type: MULTIPOLYLINE
## Dimension: XY
## Bounding box: xmin: -5.4167 ymin: 9.4295 xmax: 2.4055 ymax: 15.0652
## Geodetic CRS: +proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs
## # A tibble: 14 x 3
##   NAME_1           evenement_region           geometry
##   <chr>             <int>                  <MULTIPOLYLINE [°]>
## 1 Boucle du Mouhoun    722 (((-2.7764 13.4846), (-2.6968 13.3731), (~
## 2 Cascades            270 (((-3.7285 10.6889), (-4.0859 10.7518), (~
## 3 Centre              437 (((-1.7089 12.3306), (-1.7177 12.275), (-1~
## 4 Centre-Est          367 ((-0.36 12.18), (-0.4333 12.1833), (-0.42~
## 5 Centre-Nord         1141 ((-1.8047 13.619), (-1.7222 13.683), (-1.~
## 6 Centre-Ouest         120 ((-1.7736 11.5544), (-1.4699 11.4729), (~
## 7 Centre-Sud          58 ((-0.777 11.3352), (-0.8609 11.006), (-1.~
## 8 Est                  1675 ((-0.1305 13.5634), (-0.046 13.5326), (-0.~
## 9 Haut-Bassins         264 ((-3.5553 11.1877), (-3.535 11.2982), (-3.~
## 10 Nord                866 ((-1.881 13.9771), (-1.9307 14.0003), (-2.~
## 11 Plateau-Central    26 (((-0.6672 12.6376), (-0.9333 12.3), (-1.0.~
## 12 Sahel               2333 ((-0.4658 15.0571), (-0.4596 15.0652), (~
## 13 Sud-Ouest           164 ((-2.9333 10.9333), (-3.057 10.6322), (-3.~
## 14 <NA>                23 ((-1.6281 11.0314), (-3.9027 13.4416), (1.~

#on voit qu'il y'a un une ligne qui est NA , on la supprimer
evenement_BF1 <- subset(evenement_BF, !is.na(evenement_BF$NAME_1))
print(evenement_BF1)

## Simple feature collection with 13 features and 2 fields
## Geometry type: MULTIPOLYLINE
## Dimension: XY
## Bounding box: xmin: -5.4167 ymin: 9.4295 xmax: 2.1959 ymax: 15.0652
## Geodetic CRS: +proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs
## # A tibble: 13 x 3
##   NAME_1           evenement_region           geometry
##   <chr>             <int>                  <MULTIPOLYLINE [°]>
## 1 Boucle du Mouhoun    722 (((-2.7764 13.4846), (-2.6968 13.3731), (~
## 2 Cascades            270 (((-3.7285 10.6889), (-4.0859 10.7518), (~
## 3 Centre              437 (((-1.7089 12.3306), (-1.7177 12.275), (-1.~
## 4 Centre-Est          367 ((-0.36 12.18), (-0.4333 12.1833), (-0.42.~
## 5 Centre-Nord         1141 ((-1.8047 13.619), (-1.7222 13.683), (-1.~

```

```

## 6 Centre-Ouest
## 7 Centre-Sud
## 8 Est
## 9 Haut-Bassins
## 10 Nord
## 11 Plateau-Central
## 12 Sahel
## 13 Sud-Ouest
## avec mutate pour pouvoir representer l'object spatial et de l'utiliser après dans les raster

evenement_BF_r = New_data %>%
  group_by(NAME_1) %>%
  mutate("evenement_region"=n())
print(evenement_BF_r)

## Simple feature collection with 8466 features and 17 fields
## Geometry type: POINT
## Dimension: XY
## Bounding box: xmin: -5.4167 ymin: 9.4295 xmax: 2.4055 ymax: 15.0652
## Geodetic CRS: +proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs
## # A tibble: 8,466 x 18
## # Groups:   NAME_1 [14]
##   id      date annee type      pays      geometry GID_1   GID_0 COUNTRY
##   * <chr>  <chr> <dbl> <chr> <chr>      <POINT [°]> <chr>   <chr> <chr>
## 1 BF0101~ 30-J~ 2023 Stra~ Burk~  (-0.03 14.04) BFA.12_1 BFA Burkina Faso
## 2 BF0101~ 30-J~ 2023 Stra~ Burk~ (1.7719 12.0299) BFA.8_1 BFA Burkina Faso
## 3 BF0101~ 30-J~ 2023 Expl~ Burk~ (0.5344 10.9966) BFA.8_1 BFA Burkina Faso
## 4 BF0101~ 29-J~ 2023 Expl~ Burk~ (-1.5461 13.6781) BFA.4_1 BFA Burkina Faso
## 5 BF0101~ 29-J~ 2023 Expl~ Burk~ (-1.4348 13.9068) BFA.12_1 BFA Burkina Faso
## 6 BF0101~ 29-J~ 2023 Expl~ Burk~ (-1.6974 14.3806) BFA.12_1 BFA Burkina Faso
## 7 BF0101~ 29-J~ 2023 Viol~ Burk~ (-0.0597 14.0366) BFA.12_1 BFA Burkina Faso
## 8 BF0101~ 29-J~ 2023 Viol~ Burk~ (-3.4608 12.4634) BFA.1_1 BFA Burkina Faso
## 9 BF0101~ 29-J~ 2023 Stra~ Burk~ (-0.0964 12.4636) BFA.8_1 BFA Burkina Faso
## 10 BF0101~ 29-J~ 2023 Stra~ Burk~ (0.4311 13.7677) BFA.12_1 BFA Burkina Faso
## # i 8,456 more rows
## # i 9 more variables: NAME_1 <chr>, VARNAME_1 <chr>, NL_NAME_1 <chr>,
## #   TYPE_1 <chr>, ENGTTYPE_1 <chr>, CC_1 <chr>, HASC_1 <chr>, ISO_1 <chr>,
## #   evenement_region <int>

#representation
ggplot(evenement_BF1) +
  aes(fill = NAME_1, colour = evenement_region) +
  geom_sf(size = 1.2) +
  scale_fill_hue(direction = 1) +
  scale_color_gradient() +
  theme_minimal()

```



2.2 Evénements par province

```

pass <- here::here() #chemin relatif
pass_true <- paste0(pass,"/","gadm41_BFA_2.shp") #Accès a notre base de données
BF_provin<- read_sf(pass_true) #Lecture de notre base de données
str(BF_provin)

## sf [45 x 14] (S3: sf/tbl_df/tbl/data.frame)
## $ GID_2      : chr [1:45] "BFA.1.1_1" "BFA.1.2_1" "BFA.1.3_1" "BFA.1.4_1" ...
## $ GID_0      : chr [1:45] "BFA" "BFA" "BFA" "BFA" ...
## $ COUNTRY    : chr [1:45] "Burkina Faso" "Burkina Faso" "Burkina Faso" "Burkina Faso" ...
## $ GID_1      : chr [1:45] "BFA.1_1" "BFA.1_1" "BFA.1_1" "BFA.1_1" ...
## $ NAME_1     : chr [1:45] "Boucle du Mouhoun" "Boucle du Mouhoun" "Boucle du Mouhoun" "Boucle du Mouhoun" ...
## $ NL_NAME_1: chr [1:45] "NA" "NA" "NA" "NA" ...
## $ NAME_2     : chr [1:45] "Balé" "Banwa" "Kossi" "Mouhoun" ...
## $ VARNAME_2: chr [1:45] "Bale" "NA" "NA" "NA" ...
## $ NL_NAME_2: chr [1:45] "NA" "NA" "NA" "NA" ...
## $ TYPE_2     : chr [1:45] "Province" "Province" "Province" "Province" ...
## $ ENGTTYPE_2: chr [1:45] "Province" "Province" "Province" "Province" ...
## $ CC_2       : chr [1:45] "NA" "NA" "NA" "NA" ...
## $ HASC_2     : chr [1:45] "BF.BO.BA" "BF.BO.BW" "BF.BO.KS" "BF.BO.MO" ...
## $ geometry   :sfc_POLYGON of length 45; first list element: List of 1
##   ..$ : num [1:2158, 1:2] -2.75 -2.75 -2.74 -2.73 -2.73 ...
##   ..- attr(*, "class")= chr [1:3] "XY" "POLYGON" "sfg"
##   - attr(*, "sf_column")= chr "geometry"
##   - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",...: NA NA NA NA NA NA NA NA NA ...

```

```
## ..- attr(*, "names")= chr [1:13] "GID_2" "GID_0" "COUNTRY" "GID_1" ...
View(BF_provin)
```

Fusion de base BF_provin et de base BF_ACLED

```
base_provin= st_join(BF_ACLED,BF_provin)
head(base_provin)

## Simple feature collection with 6 features and 18 fields
## Geometry type: POINT
## Dimension: XY
## Bounding box: xmin: -1.6974 ymin: 10.9966 xmax: 1.7719 ymax: 14.3806
## Geodetic CRS: +proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0 +no_defs
## # A tibble: 6 x 19
##   id      date annee type   pays      geometry GID_2      GID_0 COUNTRY
##   <chr>    <chr> <dbl> <chr> <chr>      <POINT [°]> <chr>    <chr> <chr>
## 1 BF010160 30-J-  2023 Stra~ Burk~  (-0.03 14.04) BFA.12.2_1 BFA Burkina F~
## 2 BF010164 30-J-  2023 Stra~ Burk~  (1.7719 12.0299) BFA.8.5_1 BFA Burkina F~
## 3 BF010171 30-J-  2023 Expl~ Burk~  (0.5344 10.9966) BFA.8.4_1 BFA Burkina F~
## 4 BF010121 29-J-  2023 Expl~ Burk~  (-1.5461 13.6781) BFA.4.1_1 BFA Burkina F~
## 5 BF010125 29-J-  2023 Expl~ Burk~  (-1.4348 13.9068) BFA.12.3_1 BFA Burkina F~
## 6 BF010126 29-J-  2023 Expl~ Burk~  (-1.6974 14.3806) BFA.12.3_1 BFA Burkina F~
## # i 10 more variables: GID_1 <chr>, NAME_1 <chr>, NL_NAME_1 <chr>,
## # NAME_2 <chr>, VARNAME_2 <chr>, NL_NAME_2 <chr>, TYPE_2 <chr>,
## # ENGTTYPE_2 <chr>, CC_2 <chr>, HASC_2 <chr>
#Calcul des événements par region au burkina faso

#AVec summarise

evenement_provin_BF = base_provin %>%
  group_by(NAME_2)%>%
  summarise("evenement_province"=n())
print(evenement_provin_BF)

## Simple feature collection with 46 features and 2 fields
## Geometry type: MULTIPOINT
## Dimension: XY
## Bounding box: xmin: -5.4167 ymin: 9.4295 xmax: 2.4055 ymax: 15.0652
## Geodetic CRS: +proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0 +no_defs
## # A tibble: 46 x 3
##   NAME_2   evenement_province      geometry
##   <chr>           <int>      <MULTIPOINT [°]>
## 1 Balé            16  ((-2.8623 11.9062), (-2.93 11.745), (-3.26 11.~)
## 2 Bam             350 ((-1.8047 13.619), (-1.7222 13.683), (-1.702 1~)
## 3 Banwa           88  ((-3.7 12.5), (-3.6987 12.443), (-3.6383 12.48~)
## 4 Bazèga          12  ((-1.2664 11.83), (-1.4166 11.8166), (-1.602 1~)
## 5 Bougouriba      25  ((-3.3723 10.9984), (-3.1167 11.0833), (-3.249~)
## 6 Boulgou          145 ((-0.0135 11.8426), (-0.2 11.7333), (-0.2166 1~)
## 7 Boulkiemdé      31  ((-2.3667 12.25), (-1.85 12.3166), (-1.8856 12~)
## 8 Comoré           258 ((-3.7285 10.6889), (-4.0859 10.7518), (-4.133~)
## 9 Ganzourgou      13  ((-0.6672 12.6376), (-0.9333 12.3), (-1.0773 1~)
## 10 Gnagna          144 ((-0.1305 13.5634), (-0.046 13.5326), (-0.0748~
## # i 36 more rows
```

II Deuxième partie

II.1. Créer un raster avec une résolution de 0.5m x 0.5 m

```
# prendre le BF
r_crs <- CRS("+proj=merc +lon_0=0 +k=1 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs")
r_ext <- extent(BF) # les bords
r_res <- c(0.5,0.5) # resolution : 0.5 m
rast_zero <- raster(crs=r_crs, ext=r_ext, resolution=r_res)
st_crs(evenement_BF_r) <- r_crs

evenement_BF_r <- st_transform(evenement_BF_r, crs = r_crs)

BF_raster_reg <- rasterize(evenement_BF_r, rast_zero, field = "evenement_region", fun = "sum")

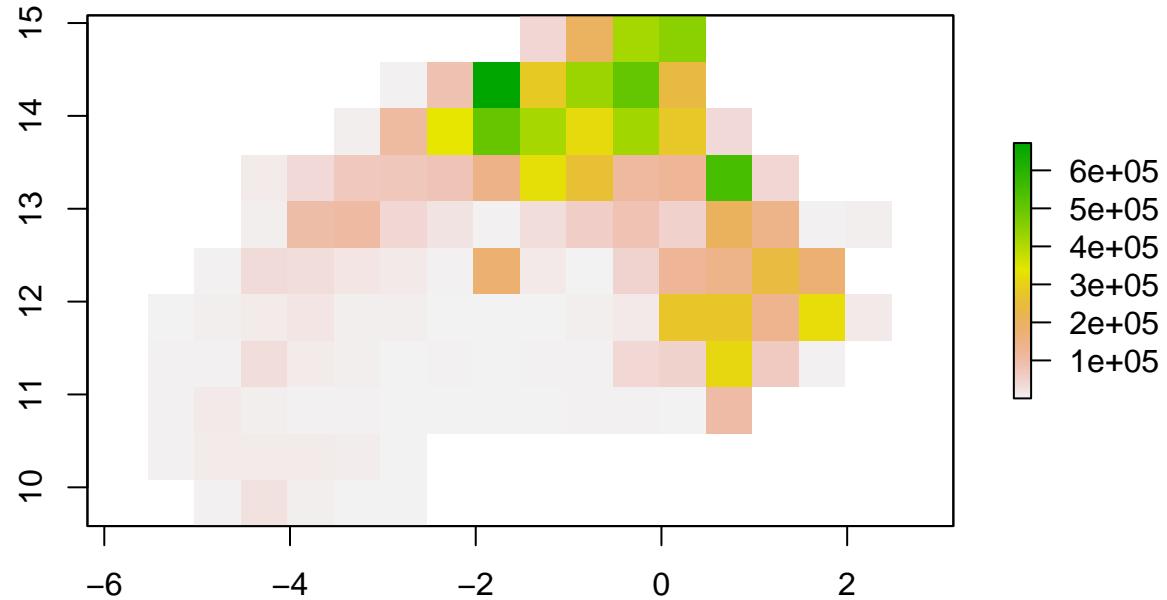
str(evenement_BF_r)

## sf [8,466 x 18] (S3: sf/grouped_df/tbl_df/tbl/data.frame)
## $ id
## $ date
## $ annee
## $ type
## $ pays
## $ geometry
## $ GID_1
## $ GID_0
## $ COUNTRY
## $ NAME_1
## $ VARNAME_1
## $ NL_NAME_1
## $ TYPE_1
## $ ENGTTYPE_1
## $ CC_1
## $ HASC_1
## $ ISO_1
## $ evenement_region: int [1:8466] 2333 1675 1675 1141 2333 2333 2333 722 1675 2333 ...
## - attr(*, "groups")= tibble [14 x 2] (S3: tbl_df/tbl/data.frame)
##   ..$ NAME_1: chr [1:14] "Boucle du Mouhoun" "Cascades" "Centre" "Centre-Est" ...
##   ..$ .rows : list<int> [1:14]
##   ... $. : int [1:722] 8 31 36 63 69 104 118 120 123 126 ...
##   ... $. : int [1:270] 28 38 70 102 139 154 157 226 265 308 ...
##   ... $. : int [1:437] 73 111 505 511 642 648 683 712 776 828 ...
##   ... $. : int [1:367] 25 48 56 62 76 89 129 131 133 135 ...
##   ... $. : int [1:1141] 4 14 17 22 23 24 29 57 58 60 ...
##   ... $. : int [1:120] 77 225 232 233 234 240 249 267 312 320 ...
##   ... $. : int [1:58] 79 939 954 956 968 1065 1189 1198 1201 1211 ...
##   ... $. : int [1:1675] 2 3 9 16 18 20 26 30 32 33 ...
##   ... $. : int [1:264] 15 27 37 43 85 92 112 169 178 185 ...
##   ... $. : int [1:866] 12 13 19 39 41 47 50 51 52 66 ...
##   ... $. : int [1:26] 1796 1937 3828 4308 5174 5179 6574 6625 7170 7314 ...
##   ... $. : int [1:2333] 1 5 6 7 10 11 21 34 44 45 ...
##   ... $. : int [1:164] 90 95 227 538 742 859 863 1075 1113 1122 ...
##   ... $. : int [1:23] 744 848 872 2454 2503 2819 2883 3560 4189 4201 ...
```

```

## ...@ ptype: int(0)
## ..- attr(*, ".drop")= logi TRUE
## - attr(*, "sf_column")= chr "geometry"
## - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",... NA ...
## ..- attr(*, "names")= chr [1:17] "id" "date" "annee" "type" ...
plot(BF_raster_reg)

```

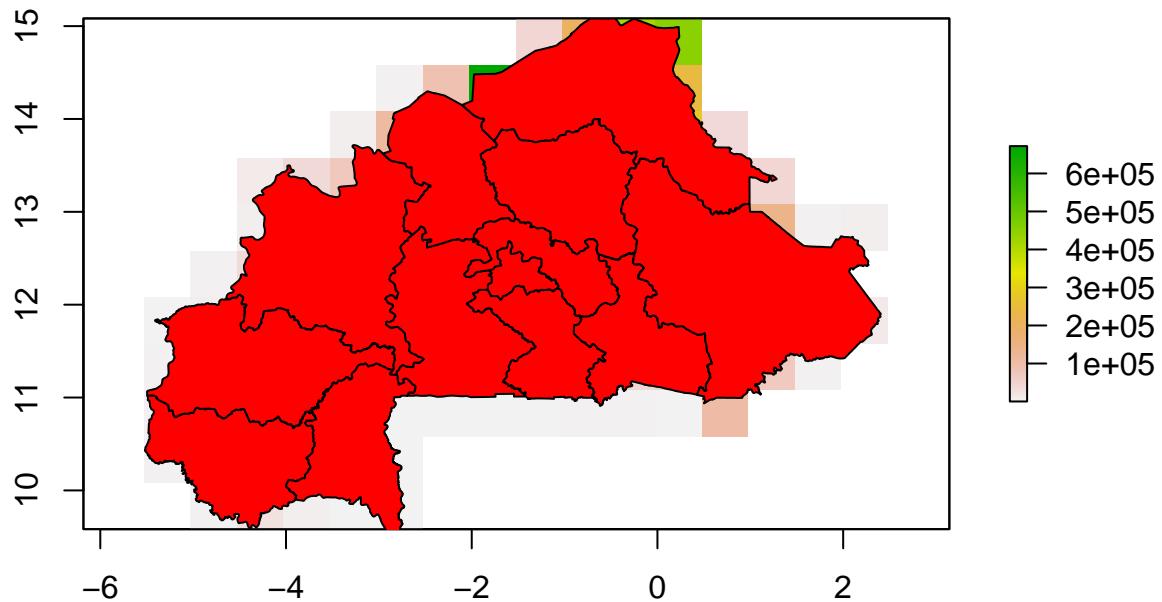


```

plot(BF_raster_reg, main = "Superposition Couche Raster et Couche Vecteur")
# Plot la couche vecteur par-dessus la couche raster
plot(BF, add = TRUE, col = "red", pch = 20)

```

Superposition Couche Raster et Couche Vecteur



II.2. creer un autre raster qui categorise le nombre d'événements

II.2.1 Categorisation du prof

```
# Categorisation des raster

# Définir les ruptures pour la catégorisation
#breaks <- c(0, 5, 10, Inf)

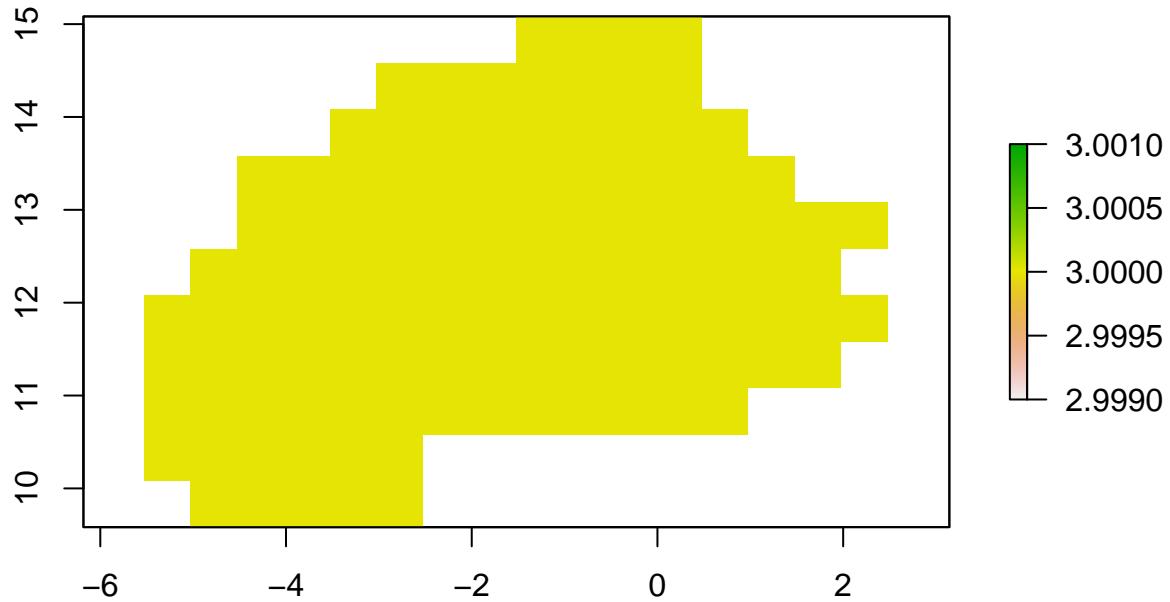
breaks <- c(0, 5, 10, Inf)

# Créer des catégories en fonction du nombre d'événements
raster_categories <- cut(getValues(BF_raster_reg), breaks = breaks, labels = c(1, 2, 3), include.lowest = TRUE)

# Créer une nouvelle couche raster avec les catégories
BF_raster_categories <- BF_raster_reg # Créer une copie de la couche raster
values(BF_raster_categories) <- as.numeric(raster_categories) # Définir les valeurs

# Plot la nouvelle couche raster avec catégories
plot(BF_raster_categories, main = "Raster avec Catégories")
```

Raster avec Catégories



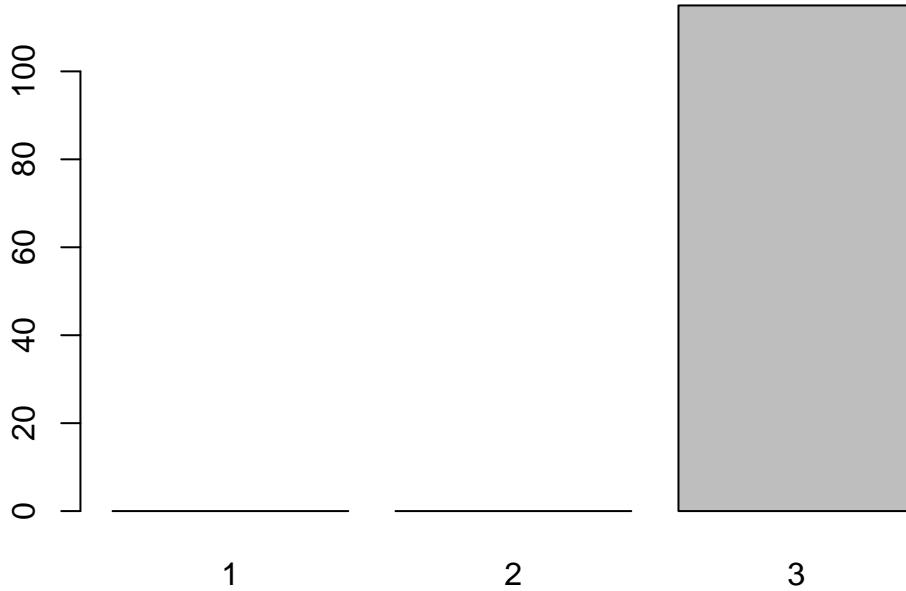
```
str(BF_raster_categories)
```

```
## Formal class 'RasterLayer' [package "raster"] with 13 slots
##  ..@ file      :Formal class '.RasterFile' [package "raster"] with 13 slots
##  ... . . . @ name      : chr ""
##  ... . . . @ datanotation: chr "FLT4S"
##  ... . . . @ byteorder   : chr "little"
##  ... . . . @ nodatavalue : num -Inf
##  ... . . . @ NAchanged   : logi FALSE
##  ... . . . @ nbands      : int 1
##  ... . . . @ bandorder   : chr "BIL"
##  ... . . . @ offset       : int 0
##  ... . . . @ toptobottom : logi TRUE
##  ... . . . @ blockrows    : int 0
##  ... . . . @ blockcols    : int 0
##  ... . . . @ driver       : chr ""
##  ... . . . @ open         : logi FALSE
##  ..@ data      :Formal class '.SingleLayerData' [package "raster"] with 13 slots
##  ... . . . @ values      : num [1:176] NA NA NA NA NA NA NA NA 3 3 ...
##  ... . . . @ offset       : num 0
##  ... . . . @ gain         : num 1
##  ... . . . @ inmemory     : logi TRUE
##  ... . . . @ fromdisk     : logi FALSE
##  ... . . . @ isfactor     : logi FALSE
##  ... . . . @ attributes   : list()
##  ... . . . @ haveminmax: logi TRUE
```

```

## ... .@ min : num 3
## ... .@ max : num 3
## ... .@ band : int 1
## ... .@ unit : chr ""
## ... .@ names : chr ""
## ..@ legend :Formal class '.RasterLegend' [package "raster"] with 5 slots
## ... .@ type : chr(0)
## ... .@ values : logi(0)
## ... .@ color : logi(0)
## ... .@ names : logi(0)
## ... .@ colortable: logi(0)
## ... @ title : chr(0)
## ..@ extent :Formal class 'Extent' [package "raster"] with 4 slots
## ... .@ xmin: num -5.52
## ... .@ xmax: num 2.48
## ... .@ ymin: num 9.58
## ... .@ ymax: num 15.1
## ... @ rotated : logi FALSE
## ..@ rotation:Formal class '.Rotation' [package "raster"] with 2 slots
## ... .@ geotrans: num(0)
## ... .@ transfun:function ()
## ... @ ncols : int 16
## ... @ nrows : int 11
## ..@ crs :Formal class 'CRS' [package "sp"] with 1 slot
## ... .@ projargs: chr NA
## ... @ srs : chr "PROJCRS[\"unknown\",\n      BASEGEOGCRS[\"unknown\",\n      DATUM[\"World Geode
# Plot les categories du raster
plot(raster_categories, main = "categorie")
```

categorie)



```
valeur = getValues(BF_raster_categories)
print(valeur)

## [1] NA NA NA NA NA NA NA NA NA 3 3 3 3 NA NA NA NA NA NA NA NA 3 3 3 3
## [26] 3 3 3 NA NA NA NA NA NA NA NA NA 3 3 3 3 3 3 3 3 3 3 3 3 3 NA NA NA NA NA
## [51] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [76] 3 3 3 3 3 NA 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [101] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [126] 3 3 NA 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [151] NA 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [176] NA
```

II.2.1 Categorisation differente de celle du prof

```
# Categorisation des raster

# Définir les ruptures pour la catégorisation
#breaks <- c(0, 5, 10, Inf)

breaks <- c(0, 1000, 2000, Inf)

# Créer des catégories en fonction du nombre d'événements
raster_categories <- cut(getValues(BF_raster_reg), breaks = breaks, labels = c(1, 2, 3), include.lowest = TRUE)

# Créer une nouvelle couche raster avec les catégories
BF_raster_categories <- BF_raster_reg # Créer une copie de la couche raster
```

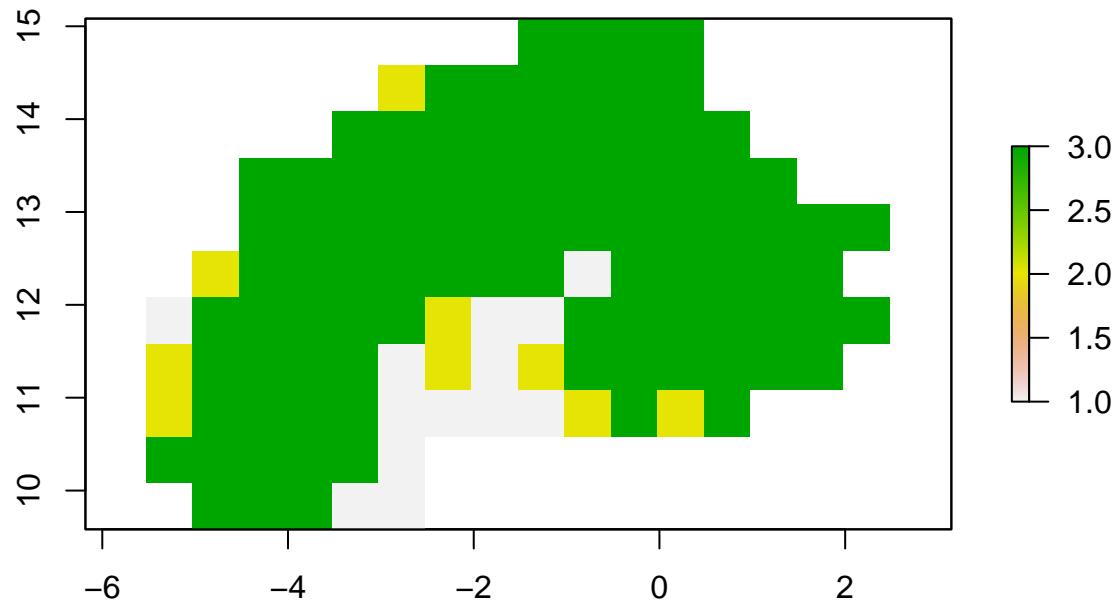
```

values(BF_raster_categories) <- as.numeric(raster_categories) # Définir les valeurs

# Plot la nouvelle couche raster avec catégories
plot(BF_raster_categories, main = "Raster avec Catégories")

```

Raster avec Catégories



```
str(BF_raster_categories)
```

```

## Formal class 'RasterLayer' [package "raster"] with 13 slots
##  ..@ file    :Formal class '.RasterFile' [package "raster"] with 13 slots
##  ... . . . @ name      : chr ""
##  ... . . . @ datanotation: chr "FLT4S"
##  ... . . . @ byteorder  : chr "little"
##  ... . . . @ nodatavalue: num -Inf
##  ... . . . @ NAchanged   : logi FALSE
##  ... . . . @ nbands     : int 1
##  ... . . . @ bandorder   : chr "BIL"
##  ... . . . @ offset      : int 0
##  ... . . . @ toptobottom : logi TRUE
##  ... . . . @ blockrows    : int 0
##  ... . . . @ blockcols    : int 0
##  ... . . . @ driver      : chr ""
##  ... . . . @ open        : logi FALSE
##  ..@ data    :Formal class '.SingleLayerData' [package "raster"] with 13 slots
##  ... . . . @ values     : num [1:176] NA NA NA NA NA NA NA NA 3 3 ...
##  ... . . . @ offset      : num 0
##  ... . . . @ gain       : num 1

```

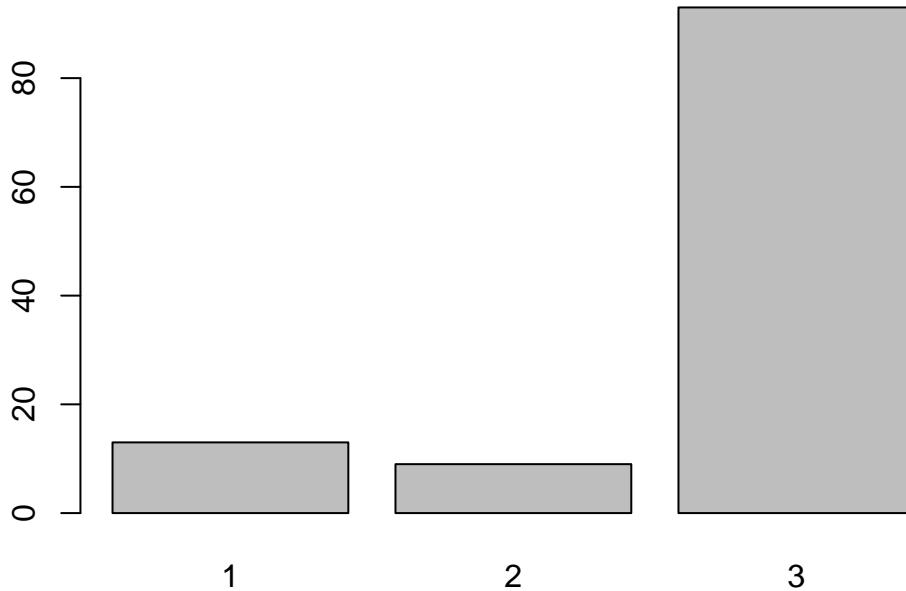
```

## ... . . . .@ inmemory : logi TRUE
## ... . . . .@ fromdisk : logi FALSE
## ... . . . .@ isfactor : logi FALSE
## ... . . . .@ attributes: list()
## ... . . . .@ haveminmax: logi TRUE
## ... . . . .@ min : num 1
## ... . . . .@ max : num 3
## ... . . . .@ band : int 1
## ... . . . .@ unit : chr ""
## ... . . . .@ names : chr ""
## ..@ legend :Formal class '.RasterLegend' [package "raster"] with 5 slots
## ... . . . .@ type : chr(0)
## ... . . . .@ values : logi(0)
## ... . . . .@ color : logi(0)
## ... . . . .@ names : logi(0)
## ... . . . .@ colortable: logi(0)
## ..@ title : chr(0)
## ..@ extent :Formal class 'Extent' [package "raster"] with 4 slots
## ... . . . .@ xmin: num -5.52
## ... . . . .@ xmax: num 2.48
## ... . . . .@ ymin: num 9.58
## ... . . . .@ ymax: num 15.1
## ..@ rotated : logi FALSE
## ..@ rotation:Formal class '.Rotation' [package "raster"] with 2 slots
## ... . . . .@ geotrans: num(0)
## ... . . . .@ transfun:function ()
## ..@ ncols : int 16
## ..@ nrows : int 11
## ..@ crs :Formal class 'CRS' [package "sp"] with 1 slot
## ... . . . .@ projargs: chr NA
## ... . . . .@ srs : chr "PROJCRS[\"unknown\",\n      BASEGEOGCRS[\"unknown\",\n      DATUM[\"World Geode"
## ..@ history : list()
## ..@ z : list()

# Plot the categorized raster
plot(raster_categories, main = "Categorized Events (10 km x 10 km)")

```

Categorized Events (10 km x 10 km)



```
valeur = getValues(BF_raster_categories)
print(valeur)

## [1] NA NA NA NA NA NA NA NA NA 3 3 3 3 NA NA NA NA NA NA NA NA NA 2 3 3 3
## [26] 3 3 3 NA NA NA NA NA NA NA NA 3 3 3 3 3 3 3 3 3 3 3 3 3 NA NA NA NA NA
## [51] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [76] 3 3 3 3 3 3 NA 2 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [101] 3 3 2 1 1 3 3 3 3 3 3 3 3 3 2 3 3 3 3 1 2 1 2 1 2 3 3 3 3 3 3 3 3
## [126] 3 3 NA 2 3 3 3 3 1 1 1 1 2 3 2 3 3 3 3 1 1 NA NA NA NA NA NA NA NA NA
## [151] NA 3 3 3 1 1 NA NA NA NA NA NA NA NA NA NA
## [176] NA
```

II.3 BONUS

Le bonus consiste à donner les limites du travail qui a été donné et autre

II.3.1 Resolution 10km*10km

La projection mercator on a une resolution en mètre donc on peut convertir km en m.

```
library(raster)
library(sf)

# Définir la projection cartographique avec une résolution en mètres
r_crs_bis <- CRS("+proj=merc +lon_0=0 +k=1 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs")

# Obtenir l'extent du raster à partir de BF
```

```

r_ext_bis <- extent(BF)

# Définir une résolution de 10 km (10 000 mètres)
r_res_bis <- c(10000, 10000)

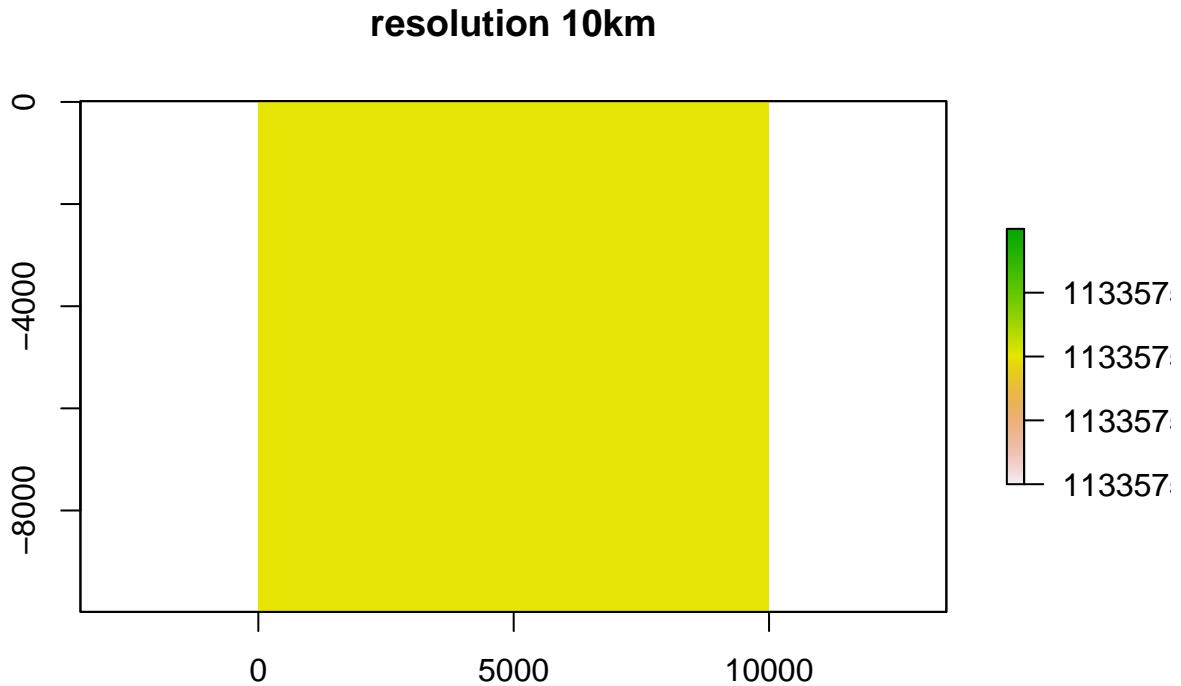
# Créer le raster avec la nouvelle résolution
rast_zero_bis <- raster(crs = r_crs_bis, ext = r_ext_bis, resolution = r_res_bis)

# Transformer les données du vecteur dans la projection cartographique
evenement_BF_r <- st_transform(evenement_BF_r, crs = r_crs_bis)

# Rasterize avec la nouvelle résolution
BF_raster_reg1 <- rasterize(evenement_BF_r, rast_zero_bis, field = "evenement_region", fun = "sum")

# Afficher le résultat
plot(BF_raster_reg1, main = "resolution 10km")

```



Mais On remarque cette resolution est trop pour le Burkina , ce qui rend l'information vague . Dons dans la suite on va changé de resolution pour pouvoir rendre l'information lisible

II.3.2 Carte du G5_sahel

Nous n'avons pas pu creer les raster pour les pays du G5 sahel du fait du volume des données

```

# Créer un raster avec une résolution de 10 km x 10 km
resolut <- 1 # en mètres

```

```

G5_crs <- CRS("+proj=sinu +lon_0=0 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m")
raster_layer <- raster(crs = G5_crs, extent(Pays_ACLED), resolution = resolut)
st_crs(New_base) <- G5_crs

image(raster_layer)
# Remplir le raster avec le nombre d'événements par cellule
raster_count <- rasterize(New_base, raster_layer, field = "pays", fun=n())

# Afficher les informations sur le raster créé
print(raster_count)

# Visualiser le raster
plot(raster_count, main = "Nombre d'événements par cellule (10 km x 10 km)")

# créer un autre raster qui categorise le nombre d'événements

# Assuming 'raster_count' is the raster layer with the number of events per cell

# Define breaks for categorization
breaks <- c(0, 5, 10, Inf)

# Create categories based on the number of events
raster_categories <- cut(getValues(raster_count), breaks = breaks, labels = c(1, 2, 3), include.lowest = TRUE)

# Create a new raster layer with categories
raster_categories <- raster_count
values(raster_categories) <- raster_categories

# Plot the categorized raster
plot(raster_categories, main = "Categorized Events (10 km x 10 km)")

```

II.3.3 Matrice de voisinage

```

#install.packages("spdep")

# Charger les packages
library(sf)
library(spdep)

# Créer la matrice de voisinage
matrice_voisinage <- poly2nb(BF_provin, queen = TRUE)

# Afficher la matrice de voisinage
matrice_voisinage

```

II.3.3.1 Sur les provinces du Burkina Faso

```

## Neighbour list object:
## Number of regions: 45
## Number of nonzero links: 216
## Percentage nonzero weights: 10.66667

```

```

## Average number of links: 4.8
burkina.spdf2 <- raster:::getData("GADM", country = "burkina faso", level = 2)

class(burkina.spdf2)

## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"

View(burkina.spdf2)
##Representation de la carte
centre<-coordinates(burkina.spdf2)
noms2<-burkina.spdf2$NAME_2
w.nb2 <- poly2nb(burkina.spdf2, row.names = noms2, queen=TRUE)
par(oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))#defini les marge exterieur et interieur
plot(burkina.spdf2)
text(centre[,1],centre[,2],noms2,cex=.7)
title("Provinces du Burkina")

```

Provinces du Burkina



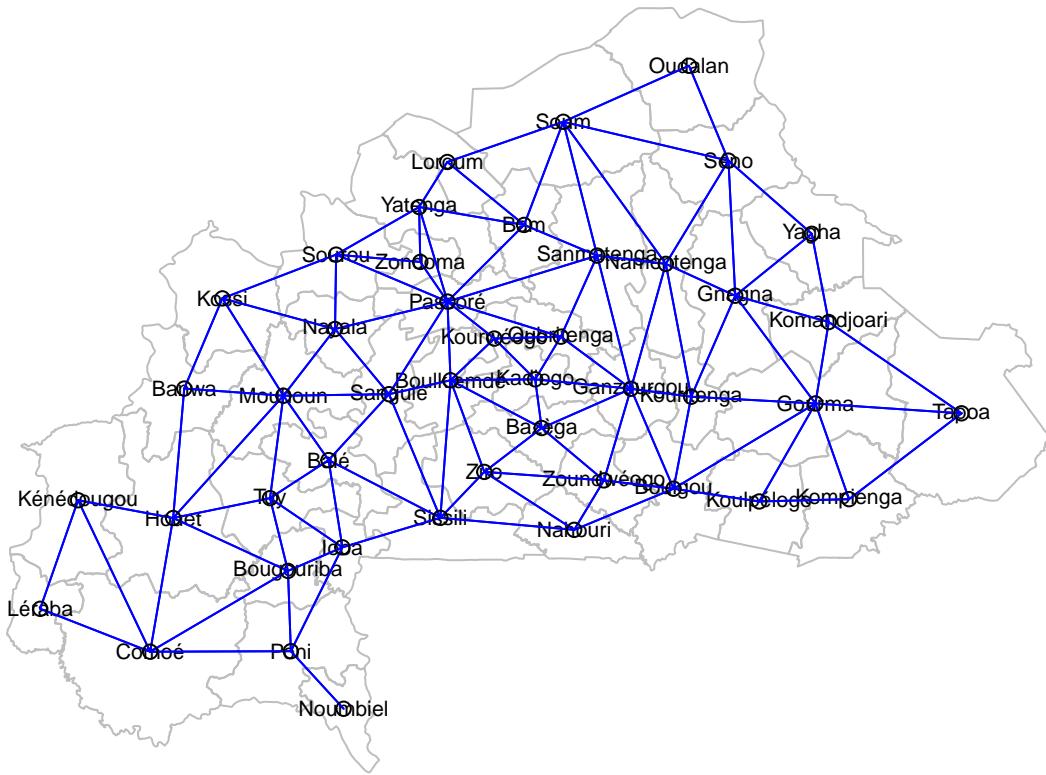
```

# les voisins

par(oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))
plot(burkina.spdf2, border="grey", axes=T)
text(centre[,1],centre[,2],noms2,cex=.7)
plot(w.nb2,centre,add=T,col="blue")
title("Matrice de voisinage des Provinces du BF")

```

Matrice de voisinage des Provinces du BF



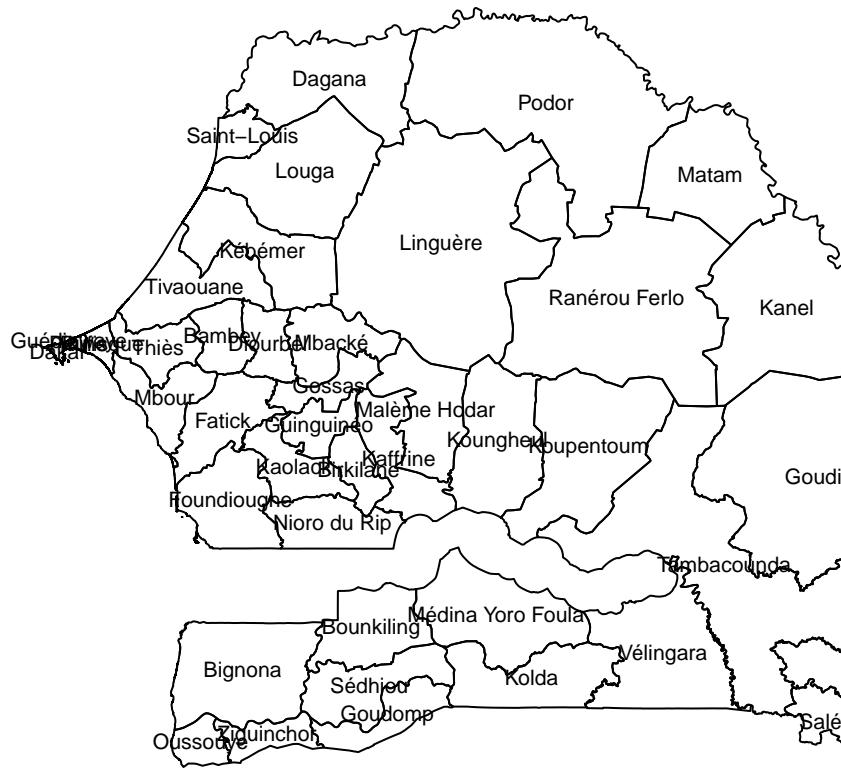
```

Senegal.spdf2 <- raster:::getData("GADM", country = "senegal", level = 2)

##Representation de la carte
centre<-coordinates(Senegal.spdf2)
noms2<-Senegal.spdf2$NAME_2
w.nb2 <- poly2nb(Senegal.spdf2, row.names = noms2, queen=TRUE)
par(oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))
plot(Senegal.spdf2)
text(centre[,1],centre[,2],noms2,cex=.7)
title("departement du senegal")

```

departement du senegal



II.3.3.1 Sur les départements du Sénégal

```
print(w.nb2)
```

```
## Neighbour list object:  
## Number of regions: 45  
## Number of nonzero links: 190  
## Percentage nonzero weights: 9.382716  
## Average number of links: 4.222222
```

Representation graphique de la matrice de voisinage du senegal

```
# les voisins

par(oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))
plot(Senegal.spdf2, border="grey", axes=T)
text(centre[,1],centre[,2],noms2,cex=.7)
plot(w.nb2,centre,add=T,col="blue")
title("Matrice de voisinage des departements du Senegal")
```

Matrice de voisinage des departements du Senegal

