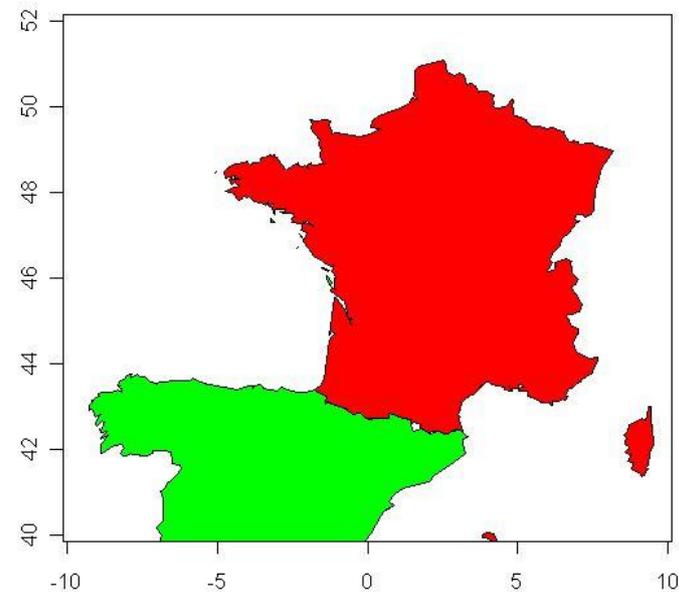
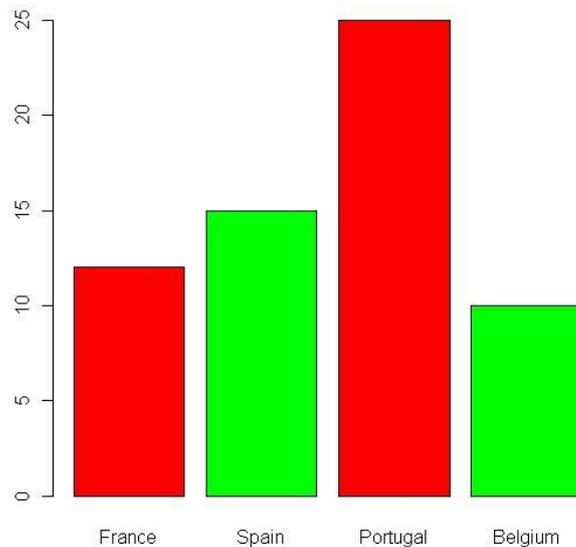


# De la cartographie autrement :

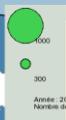
On part du principe que vous savez déjà utiliser (ou que vous avez les bases nécessaires pour apprendre à utiliser) les logiciels classiques de cartographie et les sources associées (Arcview / Shapefiles)

L'idée est de vous faire comprendre que faire un graphique à partir de données est aussi facile quand ce graphique est un histogramme que quand c'est une carte.



1

Jérôme Guillon  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

## Comment (faire des choses différentes) :

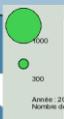
Le logiciel  avec les librairies maps, mapdata, maptools ...côté outil.

La connexion à une base de données pour ce qui est de l'acquisition de données.

## Pourquoi (faire des choses différentes) :

Pour avoir le choix.

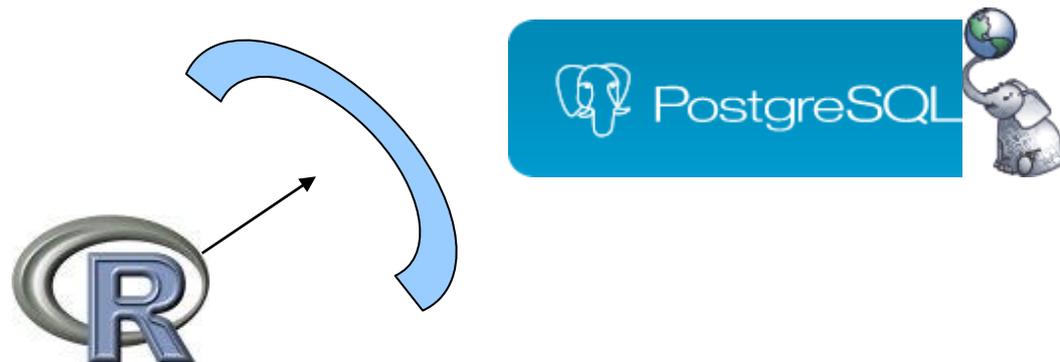
L'idée n'est pas de désigner un outil mieux qu'un autre. Quand on a le choix des outils, on peut au gré des cas d'études choisir l'une ou l'autre des solutions. C'est l'absence de choix qui est limitant



### 3 Périmètre de ce TD

On va tourner autour de :

SHP



On va chercher à regarder, atteindre :

Utilisation de différentes sources de données

Ouverture d'esprit sur des techniques moins

Possibilité de meilleur diffusion grâce au libre



3

Jérôme Guillon  
Module SIG  
2011-2012



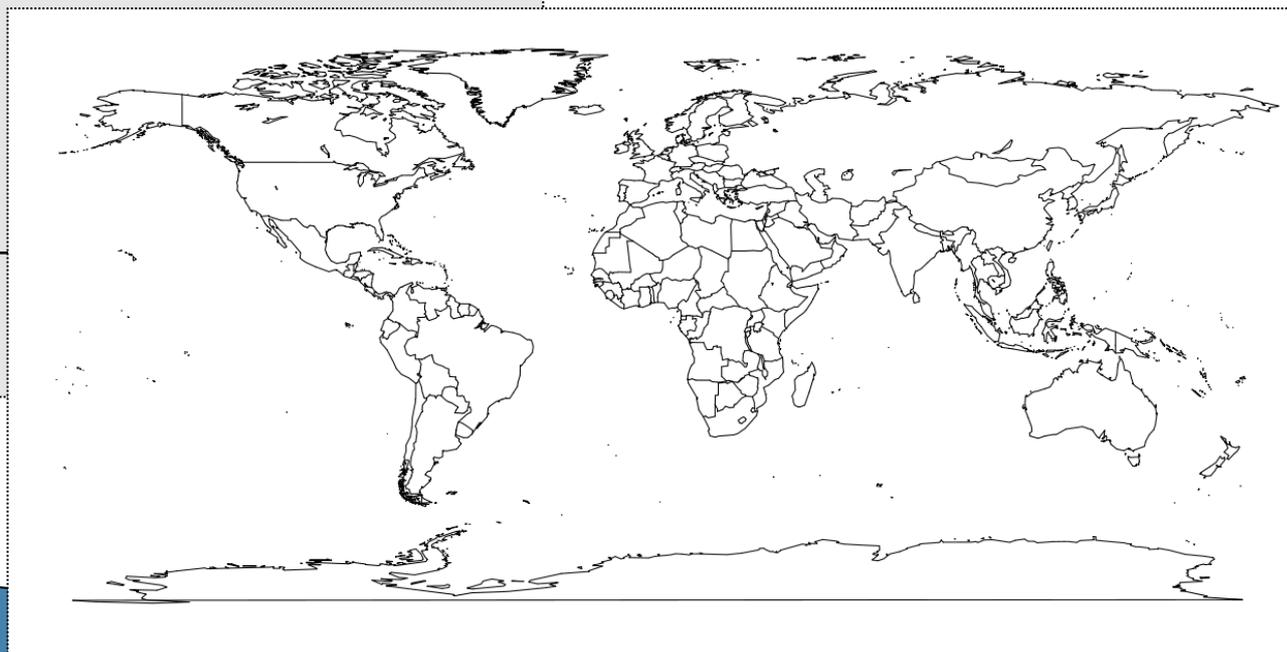
De la cartographie autrement, comment, pourquoi ?



```
library(maps)
```

→ On charge des fonctions d'accès à des bases de données des frontières mondiales + des villes + des rivières.  
Plus des données plus fines sur la France, les USA ...cf doc

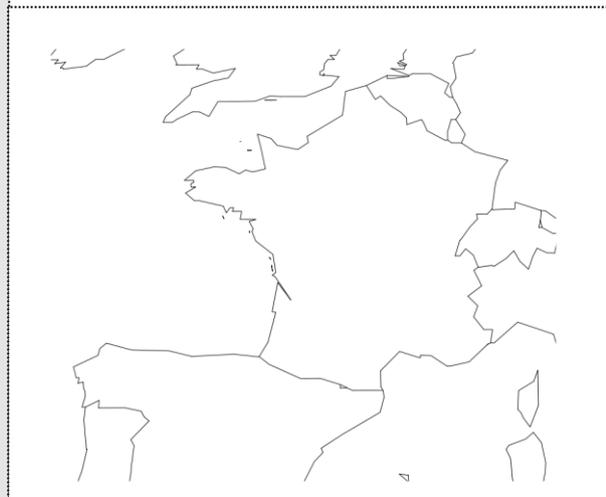
```
map()  
map('world')
```



4  la librairie maps, on commence simple

```
map(xlim=range(-10,10),ylim=range(40,52))
```

On définit l'emprise de la carte, on peut tester les paramètres **fill=T** et **col=2**



5

```
library(mapdata)  
map('worldHires',xlim=range(-10,10),  
ylim=range(40,52))
```

Accès à la base worldHires, trait plus fin.



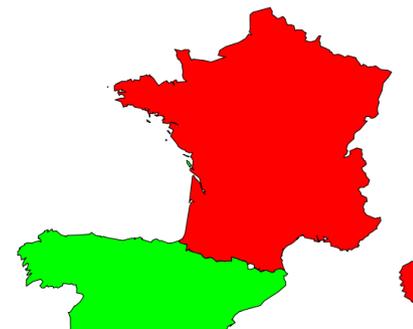
```
map('worldHires',region=c('France','Spain'),  
xlim=range(-10,10),ylim=range(40,52))
```

```
map('worldHires',region=c('France','Spain'),  
xlim=range(-10,10),ylim=range(40,52),  
col=c('red','green'),fill=T)
```

On donne le nom des polygones que l'on veut afficher.



On définit les couleurs pour les polygones



6

Jérôme Guittor  
Module SIG  
2011-2012



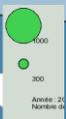
De la cartographie autrement, comment, pourquoi ?

```
captures<-  
(cbind(c('France','Spain','Portugal','Belgium'),as.numeric(c(12,15,25,10))))
```

Il nous faut maintenant définir une gamme de couleur apte à représenter les valeurs attribuées à chaque pays.

7

Jérôme Guillon  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

**##Faire une palette de couleur****library(RColorBrewer)****library(classInt)****display.brewer.all(n=NULL, type="all", select=NULL, exact.n=TRUE)****nb\_coul<-3****class <- classIntervals(as.numeric(captures[,2]), nb\_coul,  
style="pretty")****colcode <- findColours(class, brewer.pal(nb\_coul,"Reds"),  
digits=0)****colcode****map('worldHires',region=captures[,1],exact=TRUE,xlim=range(-10,10),ylim=range(40,52),fill=T,col=colcode)****legend(-6, 52, legend=names(attr(colcode, "table")),  
fill=attr(colcode, "palette"), cex=0.75,bty="n")****map.axes()**

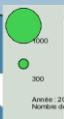
La première librairie contient un certain nombre de palettes standard.

**help(RColorBrewer)**

La seconde permet de faire des classes d'intervalle entre les différentes valeurs.

**help(classIntervals)**

8



```
##Faire une palette de couleur  
library(RColorBrewer)  
library(classInt)
```

```
nb_coul<-3  
class <- classIntervals(as.numeric(captures[,2]), nb_coul,  
style="pretty")  
colcode <- findColours(class, brewer.pal(nb_coul,"Reds"),  
digits=0)  
colcode
```

Définition des  
classes de  
couleur.

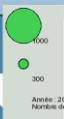
```
map('worldHires',region=captures[,1],exact=TRUE,xlim=rang  
e(-10,10),ylim=range(40,52),fill=T,col=colcode)
```

```
legend(-6, 52, legend=names(attr(colcode, "table")),  
fill=attr(colcode, "palette"), cex=0.75,bty="n")
```

```
map.axes()
```

9

Jérôme Guittor  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

```
##Faire une palette de couleur
```

```
library(RColorBrewer)
```

```
library(classInt)
```

```
nb_coul<-3
```

```
class <- classIntervals(as.numeric(captures[,2]), nb_coul,  
style="pretty")
```

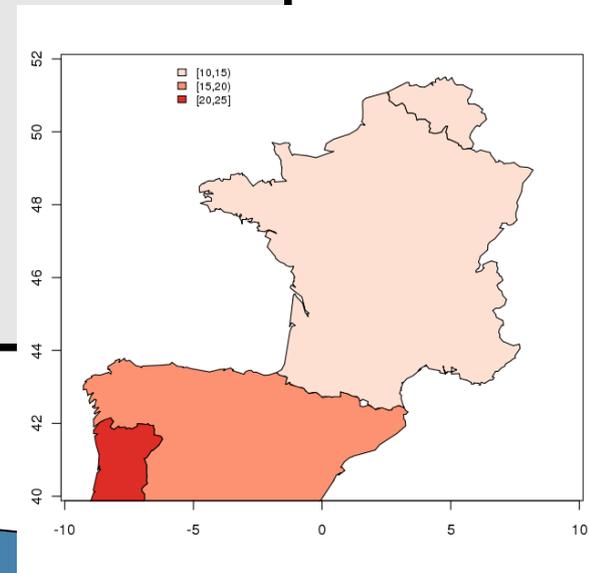
```
colcode <- findColours(class, brewer.pal(nb_coul,"Reds"),  
digits=0)
```

```
colcode
```

```
map('worldHires',region=captures[,1],exact=TRUE,xlim=rang  
e(-10,10),ylim=range(40,52),fill=T,col=colcode)
```

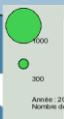
```
legend(-6, 52, legend=names(attr(colcode, "table")),  
fill=attr(colcode, "palette"), cex=0.75,bty="n")
```

```
map.axes()
```



10

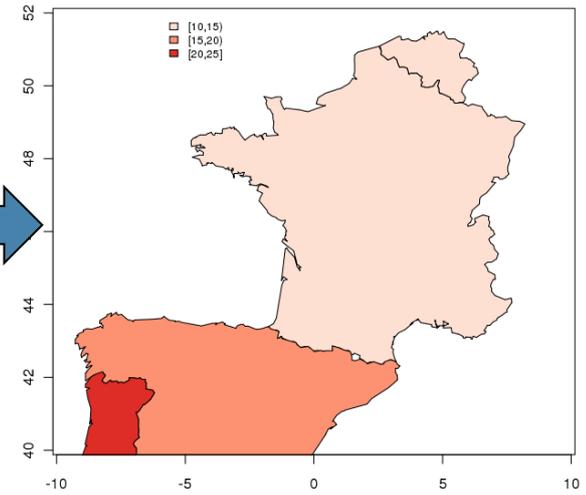
Jérôme Guittou  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

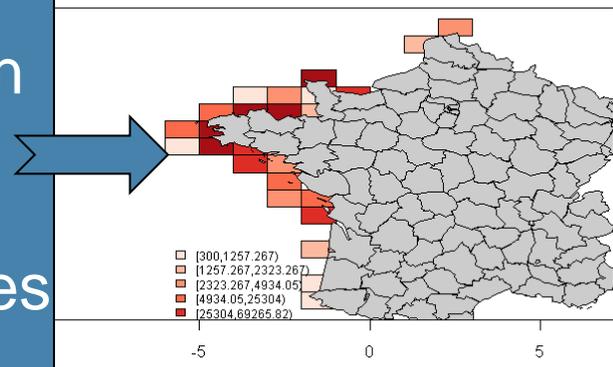
### Exemple précédent :

Formes provenant de la base de donnée  
incluse dans la librairie maps  
+  
Vecteur de captures définie sous R

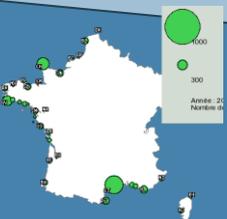


### A venir ...

Formes (rectangles CIEM) provenant d'un  
shapefile  
+  
Captures provenant d'une base de données  
à laquelle on accède par ODBC



11



La librairie **mapproj** fournit un ensemble de fonction pour manipuler et gérer les format de données géographique et notamment les fichiers ESRI shapefile.

Fonctions :

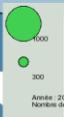
```
rect<-readShapeSpatial("logbook/IFR_SIH_RECTANGLE_P.shp")
```

La librairie **Rodbc** fournit un ensemble de fonction pour interroger les bases de données.

Fonctions :

```
channel <- odbcConnect("isih", uid="atlas", pwd="atlas")
```

```
result<-sqlQuery(channel, "select distinct substr(sect_cod,1,4) as  
sect_cod,sum(mar_qte_capturee) from asp_maree group by  
substr(sect_cod,1,4)")
```

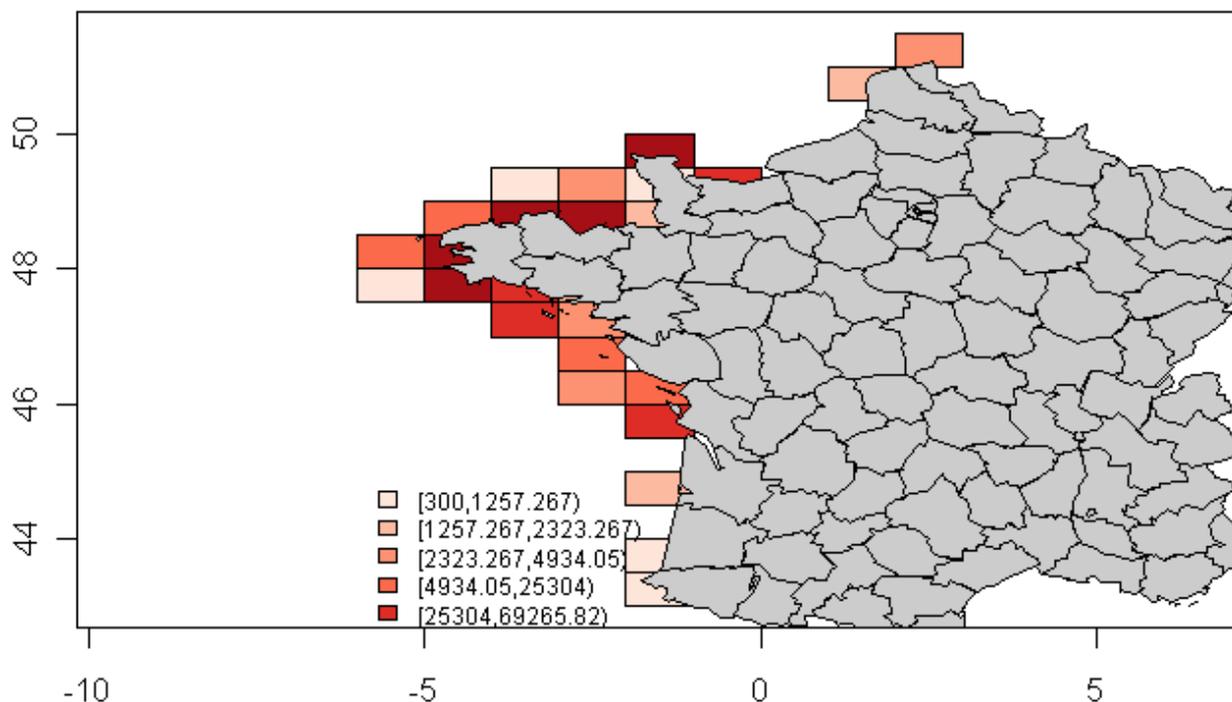


## 5 les librairies mapproj et RODBC, on vire plus compliqué

→ Un fichier Shapefile des rectangles statistiques : logbook/IFR\_SIH\_RECTANGLE\_P.shp  
Grille de rectangle statistiques de  $1/2^{\circ} \times 1^{\circ}$ , base géographique de récolte des statistiques de pêche européenne.

→ Une base de données réseau accessible à travers RODBC

→ Un trait de côte de la France avec la librairie maps.



13

Jérôme Guillon  
Module SIG  
2011-2012

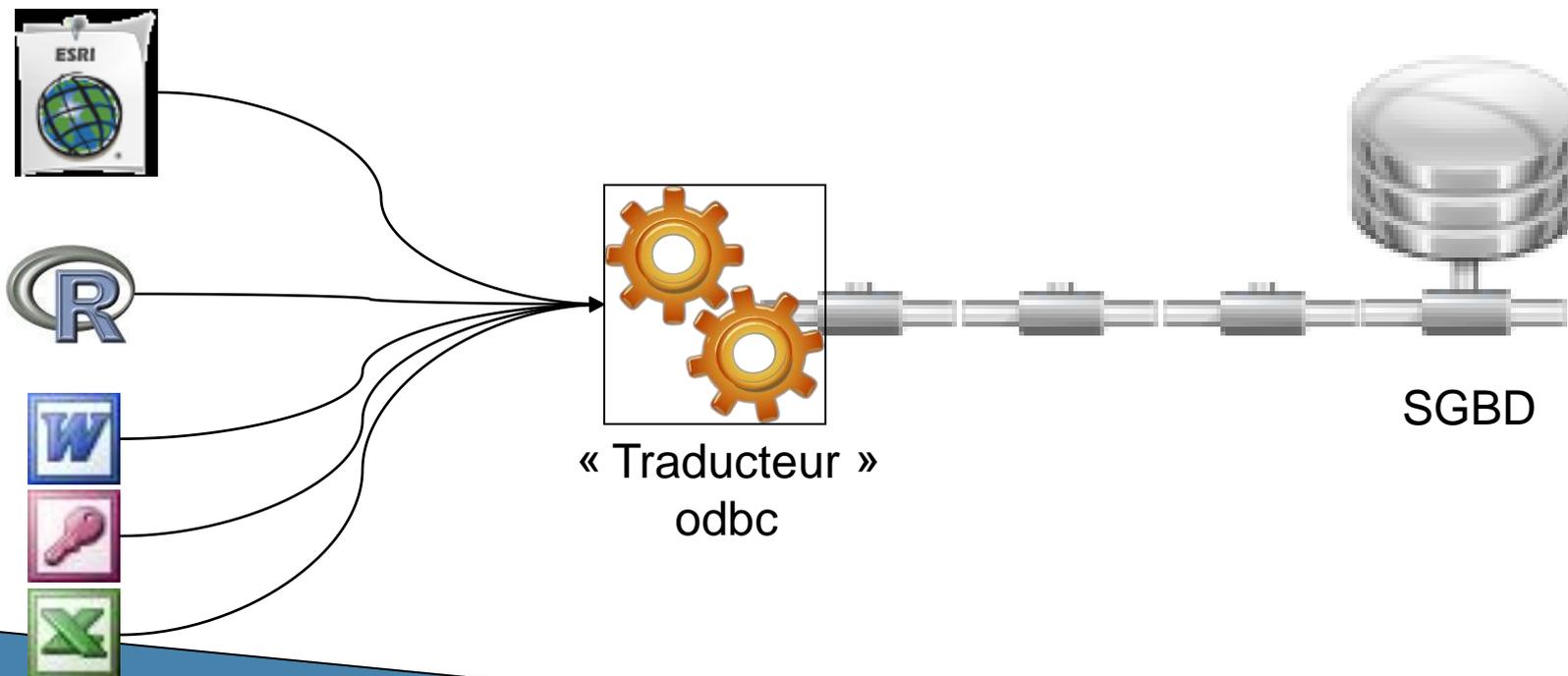


De la cartographie autrement, comment, pourquoi ?

# 5 les librairies mapprool et RODBC, on vire plus compliqué a-Open Database Connectivity, kesako ?

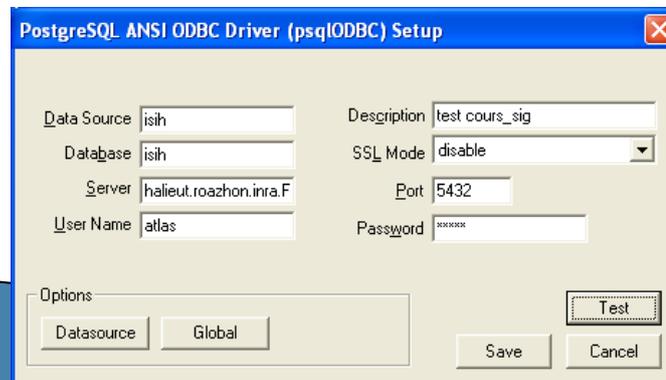
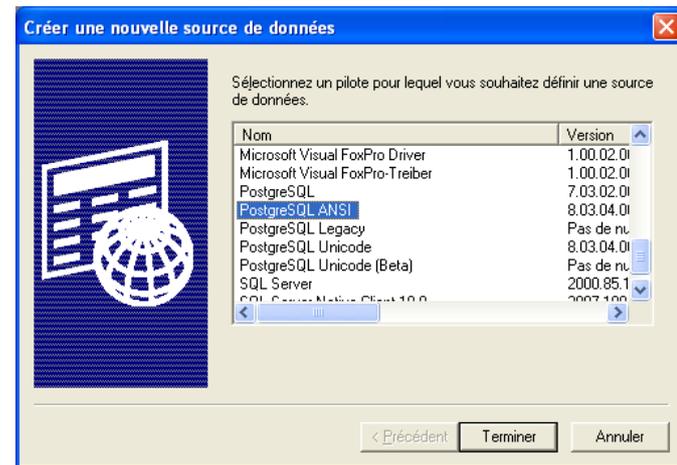
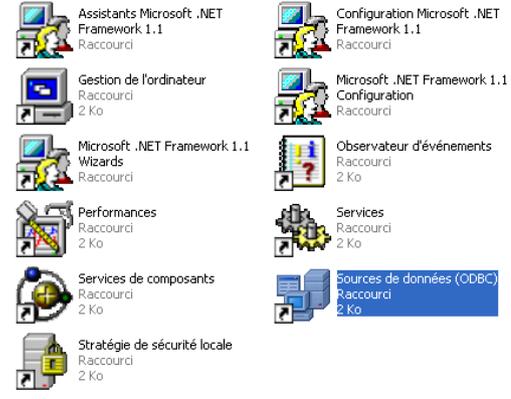
## ODBC :

C'est une méthode standard d'accès aux bases de données

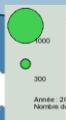


# 5 les bibliothèques mapprool et RODBC, on vire plus compliqué

## b-Mise en place connexion ODBC



De la cartographie autrement, comment, pourquoi ?



5 les librairies mapproj et RODBC, on vire plus compliqué

c-Interrogation de la base

	A	B	C	G	H	I	J	L	N	T	U
1	navs_cod	mar_annee	mar_mois	tsect_cod	sect_cod	tlieu_cod	lieu_cod	esp_cod	engin_cod	mar_qte_capturee	mar_montant_capturee
2	156162	2003	8	14	25E510	6	XBR	3213	242	18.6	82
3	156162	2003	8	14	25E510	6	XBR	3309	242	68.7	864
4	156162	2003	8	14	25E510	6	XBR	3705	242	1.1	1
5	156162	2003	8	14	25E510	6	XBR	4301	242	11.4	221
6	156162	2003	8	14	25E510	6	XBR	4304	242	22.5	435
7											36
8	<b>library(RODBC)</b>										2
9											1602
10											31
11	<b>#On déclare la base sous panneau config --&gt; outil administrations -&gt; source odbc</b>										118
12											527
13	<b>channel &lt;- odbcConnect("isih", uid="atlas", pwd="atlas")</b>										327
14											31

```

library(RODBC)

#On déclare la base sous panneau config --> outil administrations -> source odbc

channel <- odbcConnect("isih", uid="atlas", pwd="atlas")

result<-sqlQuery(channel, "select distinct substr(sect_cod,1,4) as
sect_cod,sum(mar_qte_capturee) from asp_maree group by substr(sect_cod,1,4)")
    
```

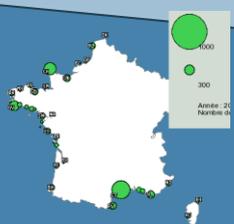
Traduction de la requête SQL :

Regroupez toutes les lignes ou il y a même code secteur (sur 4 caractères) et summez les captures correspondantes

	A	B
1	substr(sect_cod,1,4)	sum(mar_qte_capturee)
2	26E7	100984.5
3	24E7	1955.2
4	22E7	8530
5	25E5	70450.4
6	25E4	6480
7	20E8	33893.4
8	15E8	903.4
9	28E8	70007.1
10	23E7	2329.8
11	16E8	1045.5
12	21E8	5146.6
13	27E9	29643.1

16

Jérôme Guitton  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

5  les librairies mapprool et RODBC, on vire plus compliqué

c-Interrogation de la base -Pense bête SQL

**Select distinct** [Liste des champs qui nous intéressent séparés par des virgules]  
**from** [La ou les tables qui nous intéressent – incluant les liaisons entre ces tables]  
**where** [liste de conditions de sélection des lignes qui nous intéressent]  
**group by** [Liste des champs que l'on veut regrouper]

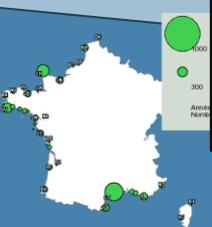
**Select distinct mar\_annee from**  
**from asp\_maree**

17

**edit(sqlQuery(channel, "select distinct mar\_annee from asp\_maree"))**  
**edit(sqlQuery(channel, "select mar\_annee from asp\_maree"))**  
**edit(sqlQuery(channel, "select distinct mar\_annee,mar\_mois from asp\_maree"))**  
**edit(sqlQuery(channel, "select distinct mar\_annee,min(mar\_mois) from asp\_maree"))** ✘

ERROR: column \"asp\_maree.mar\_annee\"  
 must appear in the GROUP BY clause or be used in an aggregate function;  
 Error while executing the query",

De la cartographie autrement, comment, pourquoi ?



5 les librairies mapproj et RODBC, on vire plus compliqué

c-Interrogation de la base -Pense bête SQL

```
edit(sqlQuery(channel, "select distinct mar_annee from asp_maree"))
edit(sqlQuery(channel, "select distinct mar_annee,mar_mois from asp_maree"))
edit(sqlQuery(channel, "select distinct mar_annee,min(mar_mois) from asp_maree"))
```

ERROR: column "asp\_maree.mar\_annee"  
must appear in the GROUP BY clause or be used in an aggregate function;\n  
Error while executing the query",

```
edit(sqlQuery(channel, "select distinct mar_annee,min(mar_mois) from asp_maree group by mar_annee"))
```

**Au départ**

mar_annee	mar_mois
2003	8
2003	9
2003	10
2004	8
2004	9
2004	10

**On regroupe les lignes qui ont même mar\_annee**

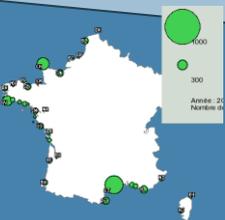
mar_annee	mar_mois
2003	8,9,10
2004	8,9,10

**Sur les mois regroupés, on applique la fonction min**

mar_annee	min(mar_mois)
2003	min(8,9,10)
2004	min(8,9,10)

**Résultat final**

mar_annee	min(mar_mois)
2003	8
2004	8



5 les librairies mapprool et RODBC, on vire plus compliqué

c-Interrogation de la base

	A	B	C	G	H	I	J	L	N	T	U
1	navs_cod	mar_annee	mar_mois	tsect_cod	sect_cod	tlieu_cod	lieu_cod	esp_cod	engin_cod	mar_qte_capturee	mar_montant_capture
2	156162	2003	8	14	25E510	6	XBR	3213	242	18.6	82
3	156162	2003	8	14	25E510	6	XBR	3309	242	68.7	864
4	156162	2003	8	14	25E510	6	XBR	3705	242	1.1	1
5	156162	2003	8	14	25E510	6	XBR	4301	242	11.4	221
6	156162	2003	8	14	25E520	6	1BR	4301	242	22.5	435
7	156162	2003	8	14	25E520	6	1BR	3213	321	8.1	36
8	156162	2003	8	14	25E520	6	1BR	3358	321	3.1	2
9	176216	2003	8	14	24E630	6	GLO	3213	320	362	1602
10	176216	2003	8	14	24E630	6	GLO	3102	511	2	31
11	176216	2003	8	14	24E630	6	GLO	3121	511	9	118
12	176216	2003	8	14	24E630	6	GLO	3202	511	130	527
13	176216	2003	8	14	24E630	6	GLO	3376	511	67	327
14	176216	2003	8	14	24E630	6	GLO	3892	511	16	31

```

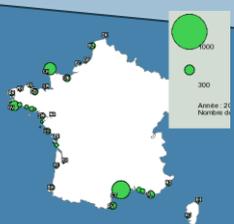
result<-sqlQuery(channel, "select distinct
substr(sect_cod,1,4) as sect_cod,
sum(mar_qte_capturee)
from asp_maree group by substr(sect_cod,1,4)
")
    
```

Je ne veux que les 4 premiers caractères du sect\_cod  
substr(monchamp,a partir de , pour n caract)

Je veux regrouper les lignes qui ont les même  
substr(sect\_cod,1,4) et sur les valeurs  
De mar\_qte\_capturee ainsi regroupées,  
je veux appliquer la fonction sum

19

Jérôme Guillon  
Module SIG  
2011-2012



Fin de l'incursion SQL.

De la cartographie autrement, comment, pourquoi ?

5  les librairies mapproj et RODBC, on vire plus compliqué

## d- Chargement du fichier de formes (rectangles)

```
library(mapproj)
rect<-readShapeSpatial("logbook/IFR_SIH_RECTANGLE_P.shp")

#Pour visualiser les rectangles

plot(rect)
map(add=T)
map.axes()

#Pour savoir a quoi correspond la structure rect

rect[rect@data$SECT_COD=="25E5",]
```

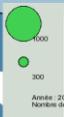
Structure polygone :

2 choses importantes dans la structure :

- 1 matrice de données rect@data, dont le code du secteur (sect\_cod)
- 1 matrice de coordonnées

20

Jérôme Guillon  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

## 5 les bibliothèques mapproj et RODBC, on vire plus compliqué e- liaison entre les 2 sources d'information

1 objet result avec les captures  
par rectangle stat

+

1 objet rect avec les informations  
géographiques sur chaque  
rectangle

1- Filtrer mes  
rectangles sur ceux où  
j'ai de la donnée

```
rect2<-rect[rect@data$SECT_COD %in% result$sect_cod,]  
#rect@data$sect_cod sont les sect_cod de mes objets geo  
#result$sect_cod sont les sect_cod de mon interrogation SQL  
#Les filtres en R : rect[ici je met un filtre,]
```

2- Ajouter à mon objet  
rect les captures de  
chaque rectangle

```
rect2@data <-  
merge(rect2@data,result,by.x='SECT_COD',by.y='sect_cod',  
all.x=TRUE,all.y=TRUE,sort=FALSE)
```

21

Jérôme Guillon  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

## 5 les librairies maptool et RODBC, on vire plus compliqué e- liaison entre les 2 sources d'information

1- Filtrer mes rectangles : je ne garde dans rect que les rectangles pour lesquels j'ai de la donnée

```
rect2<-rect[rect@data$SECT_COD %in% result$sect_cod,]
```

Données de rect (rect@data)

OBJECTID	F_AREA	F_SUBAREA	F_DIVISION	SECT_COD
6421	27	27.7	27.7.e	28E5
6110	27	27.7	27.7.a	35E5
6549	27	27.7	27.7.e	25E5
6199	27	27.7	27.7.a	33E5
6466	27	27.7	27.7.e	27E6
6243	27	27.7	27.7.f	32E6
6111	27	27.7	27.7.a	35E6
6378	27	27.7	27.7.e	29E6
6509	27	27.7	27.7.e	26E6
6333	27	27.7	27.7.e	30E6
6288	27	27.7	27.7.f	31E6
6058	27	27.7	27.7.a	36E6
6422	27	27.7	27.7.e	28E6
6789	27	27.8	27.8.d	19E2
6788	27	27.8	27.8.d	19E1
6791	27	27.8	27.8.d	19E4
6790	27	27.8	27.8.d	19E3
6749	27	27.8	27.8.d	20E4
6746	27	27.8	27.8.d	20E1
6747	27	27.8	27.8.d	20E2
6748	27	27.8	27.8.d	20E3
6707	27	27.8	27.8.d	21E3
6705	27	27.8	27.8.d	21E1
6708	27	27.8	27.8.d	21E4
6706	27	27.8	27.8.d	21E2
6667	27	27.8	27.8.d	22E4
6664	27	27.8	27.8.d	22E1
6665	27	27.8	27.8.d	22E2
6666	27	27.8	27.8.d	22E3
6624	27	27.8	27.8.d	23E2
6623	27	27.8	27.8.d	23E1

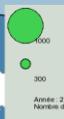
Result (captures par rectangle statistiques)

	sect_cod	sum
1	15E8	903.4
2	16E8	1045.5
3	18E8	1686.6
4	20E8	33893.4
5	21E7	2490
6	21E8	5146.6
7	22E7	8530
8	23E6	36595
9	23E7	2329.8
10	24E4	1030
11	24E5	74940.7
12	24E6	65559.4
13	24E7	1955.2
14	25E4	6480
15	25E5	70450.4
16	26E5	16625.8
17	26E6	369127.4
18	26E7	100984.5
19	26E8	1936
20	27E6	300
21	27E7	3882.5
22	27E8	1171.4
23	27E9	29643.1
24	28E8	70007.1

Ligne conservée

22

Jérôme Guillon  
Module SIG  
2011-2012



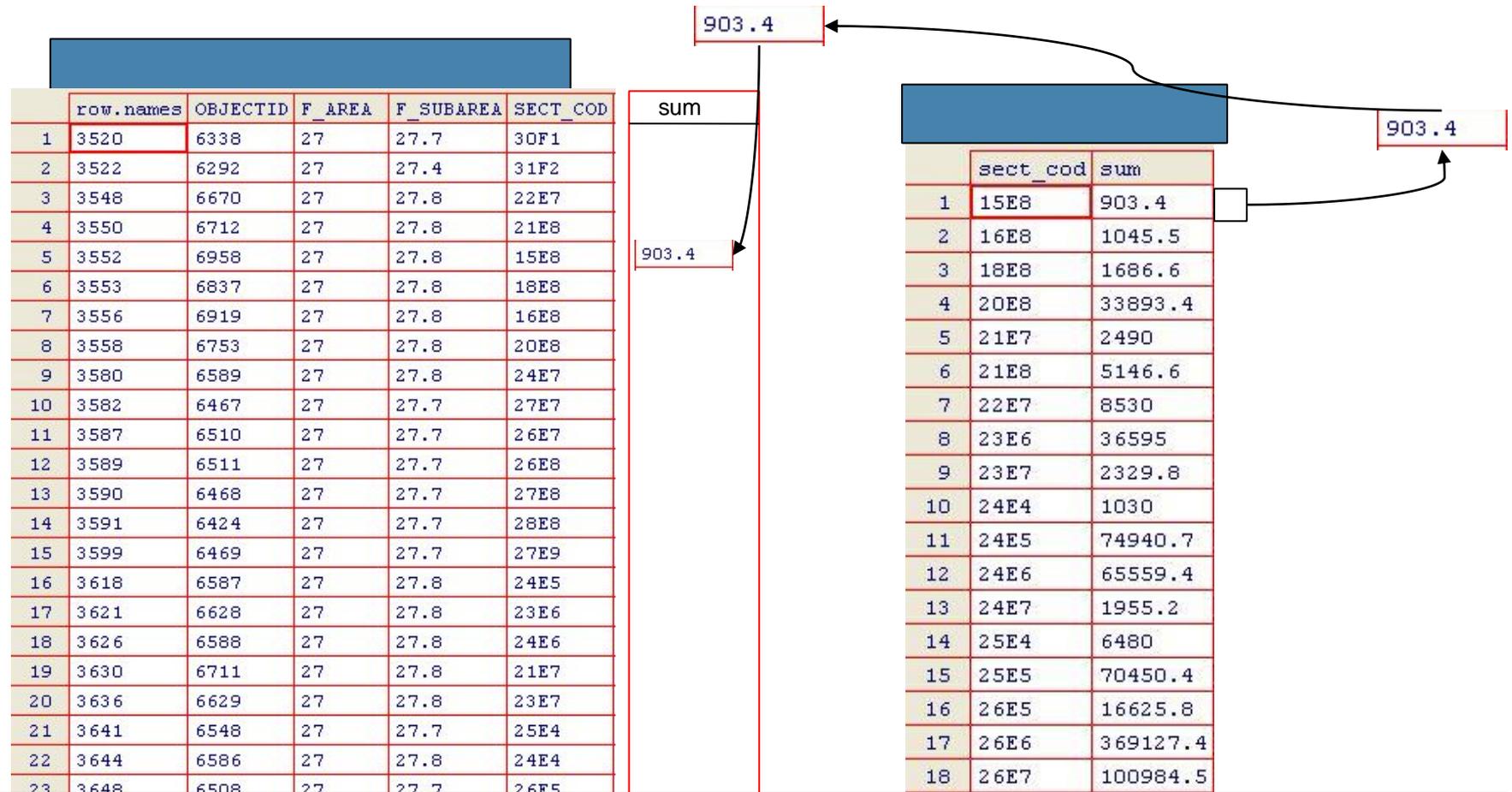
De la cartographie autrement, comment, pourquoi ?

5 les librairies mapproj et RODBC, on vire plus compliqué

e- liaison entre les 2 sources d'information

2- Ajouter à mon objet rect les captures de chaque rectangle

```
rect2@data <- merge(rect2@data,result,by.x='SECT_COD',by.y='sect_cod', all.x=TRUE,all.y=TRUE,sort=FALSE)
```



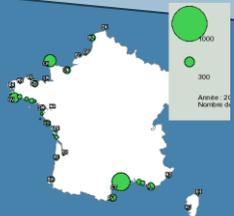
23

Jérôme Guillon  
Module SIG  
2011-2012

**Rect@data** Devient le résultat de la jointure entre les 2 structures sur la base du même champ sect\_cod

23	27E9	29643.1
24	28E8	70007.1

De la cartographie autrement, comment, pourquoi ?



5  les librairies mapproj et RODBC, on vire plus compliqué

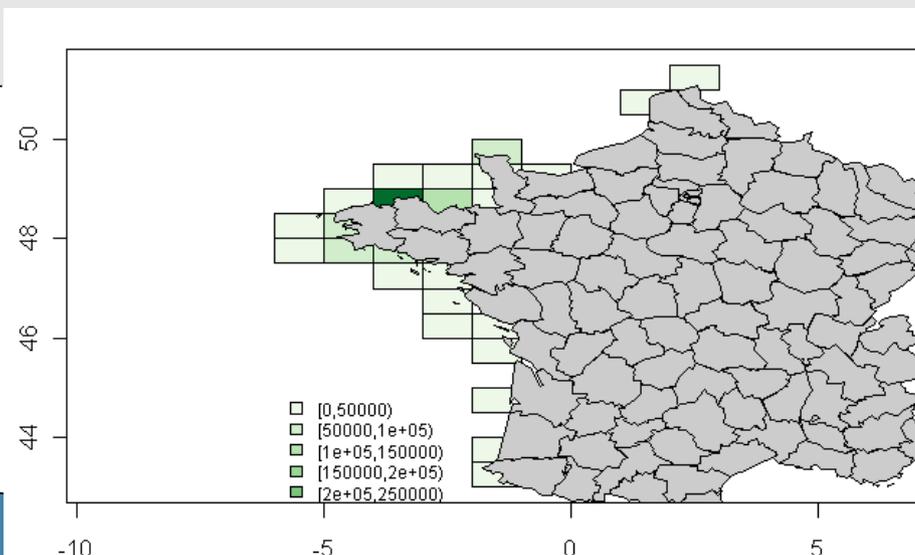
## e- Palette et affichage

```
library(RColorBrewer)
library(classInt)
```

```
class <- classIntervals(rect2@data$sum, nclr, style="pretty")
colcode <- findColours(class, brewer.pal(nclr,"Greens"), digits=0,cutlabels=TRUE)
colcode
```

```
plot(rect2,col=colcode)
legend(-6, 45, legend=names(attr(colcode, "table")), fill=attr(colcode, "palette"),
cex=0.75, bty="n")
```

```
library(maps)
map("france",add=T,col="#cccccc",fill=TRUE)
map.axes()
```



24

Jérôme Guillon  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

## 6 On fait du bis repetita

### a-b-c-d - Obtention des données, liaisons, et affichage

	A	B	C	G	H	I	J	L	N	T	U
1	navs_cod	mar_annee	mar_mois	tsect_cod	sect_cod	lieu_cod	lieu_cod	esp_cod	engin_cod	mar_qte_capturee	mar_montant_capturee
2	156162	2003	8	14	25E510	6	XBR	3213	242	18.6	82
3	156162	2003	8	14	25E510	6	XBR	3309	242	68.7	864
4	156162	2003	8	14	25E510	6	XBR	3705	242	1.1	1
5	156162	2003	8	14	25E510	6	XBR	4301	242	11.4	221
6	156162	2003	8	14	25E520	6	1BR	4301	242	22.5	435
7	156162	2003	8	14	25E520	6	1BR	3213	321	8.1	36
8	156162	2003	8	14	25E520	6	1BR	3358	321	3.1	2
9	176216	2003	8	14	24E630	6	GLO	3213	320	362	1602
10	176216	2003	8	14	24E630	6	GLO	3102	511	2	31
11	176216	2003	8	14	24E630	6	GLO	3101	511	0	118

```
result<-sqlQuery(channel, "select distinct substr(lieu_cod,2,2) as lieu_cod,sum(mar_qte_capturee) from asp_maree group by substr(lieu_cod,2,2)")
```

```
port<-readShapePoints("logbook/IFR_L_PORT_IMMAT.shp")
```

```
port2<-port[port@data$LIEU_COD %in% result$lieu_cod,]
```

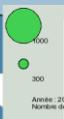
A	B
substr(lieu_cod,2,2)	sum(mar_qte_capturee)
AC	1686.6
AD	41178.5
AY	36695
BA	1948.9
BL	3699
BR	22387.9

Le fichier de forme des ports est constitué de points (1 coordonnée par objet)

On filtre sur les données que l'on a dans nos résultats

25

Jérôme Guittor  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

## 6 On fait du bis repetita

## a-b-c-d - Obtention des données, liaisons, et affichage

```
port2@data <- merge(port2@data,result, by.x='LIEU_COD',by.y='lieu_cod',  
all.x=TRUE,all.y=TRUE,sort=FALSE)
```

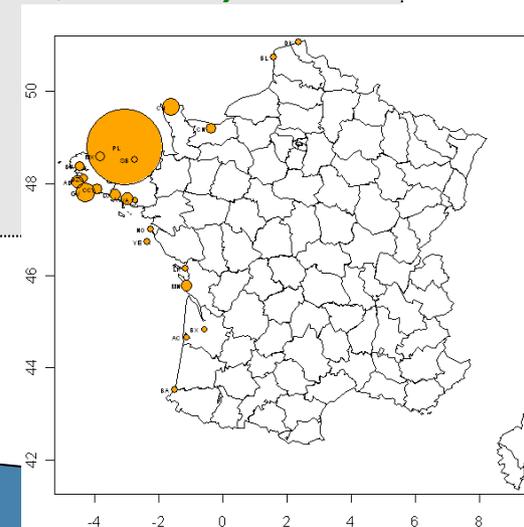
```
max.symbol.size=12  
min.symbol.size=1
```

```
plotvar <- port2@data$sum  
symbol.size <- ((plotvar-min(plotvar))/(max(plotvar)-min(plotvar))*(max.symbol.size-  
min.symbol.size)+min.symbol.size)
```

```
map("france")
```

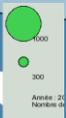
```
points(port2,pch=19, cex=symbol.size,col="orange")  
points(port2,pch=1, cex=symbol.size,col="#000000")
```

```
text(port2@coords[,1],port2@coords[,2],adj=1.6,port2$LIEU_COD,cex=0.4)  
map.axes()
```



26

Jérôme Guillon  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

Idée :

Je sais faire une carte de manière simple, puis-je automatiser la production de ce type de graphe.

Question : je veux une carte par année

Algorithme :

- Récupérer les différentes années disponible

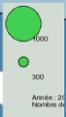
Boucle sur les années

```
{  
  Obtenir les donnée pour années i  
  Faire carte année i  
  Conserver captures totales années i  
}
```

Barplot(années)

27

Jérôme Guillon  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

## - Récupérer les différentes années disponible

```
liste<-sqlQuery(channel, "select distinct mar_annee from asp_maree")
```

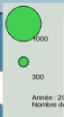
```
#Je découpe ma fenêtre en 2 lignes 2 colonnes pour pouvoir mettre 4 graphiques  
op <- par(mfrow = c(2, 2),pty = "s")
```

```
max.symbol.size=8  
min.symbol.size=1
```

```
#J'initialise un tableau pour mes sommes par années  
somme<-array()
```

```
#Je charge mon fichier de forme des ports.
```

```
port<-readShapePoints("logbook/IFR_L_PORT_IMMAT.shp")
```



## - Boucler sur les années

```
for (l in 1:(length(liste)+1))
```

```
{
```

```
result<-sqlQuery(channel,paste("select distinct substr(lieu_cod,2,2) as  
lieu_cod,sum(mar_qte_capturee) from asp_maree  
where mar_annee=",liste[l,]," group by substr(lieu_cod,2,2)"))
```

```
port_tmp<-port[port@data$LIEU_COD %in% result$lieu_cod,]
```

```
port_tmp@data <- merge(port_tmp@data,result, by.x='LIEU_COD',by.y='lieu_cod',  
all.x=TRUE,all.y=TRUE,sort=FALSE)
```

```
map("france")
```

```
plotvar <- port_tmp@data$sum
```

```
symbol.size <- ((plotvar-min(plotvar))/(max(plotvar)-min(plotvar))*(max.symbol.size-  
min.symbol.size)+min.symbol.size)
```

```
points(port_tmp,pch=19, cex=symbol.size,col="orange")
```

```
points(port_tmp,pch=1, cex=symbol.size,col="#000000")
```

```
title(paste("Captures Francaise pour ",liste[l,]))
```

```
map.axes()
```

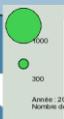
```
somme[paste("",liste[l,],"")]<-sum(port_tmp@data$sum)
```

```
}
```

- Obtenir les données pour l'année N (liste[i,])

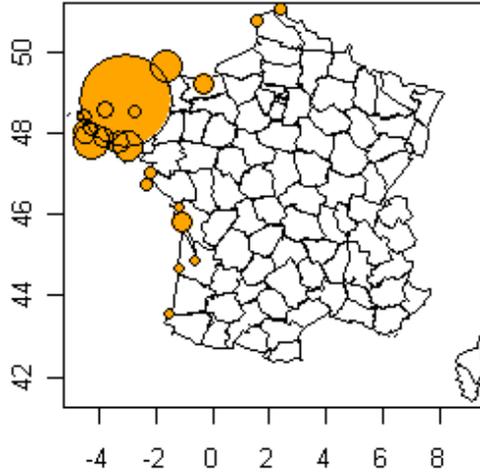
- Faire carte année N (année liste[i,])

- Conserver la somme de l'année N

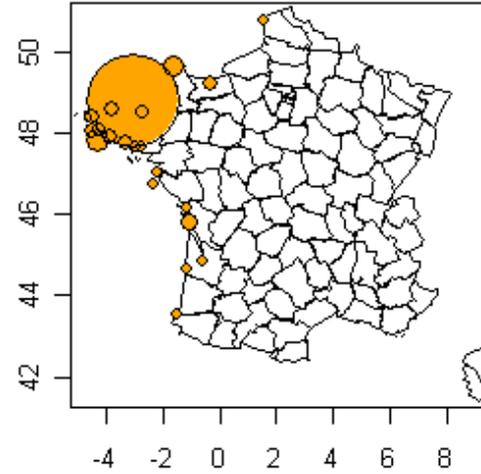


# 7 Automatisation de la procédure

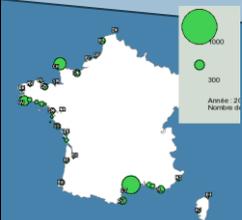
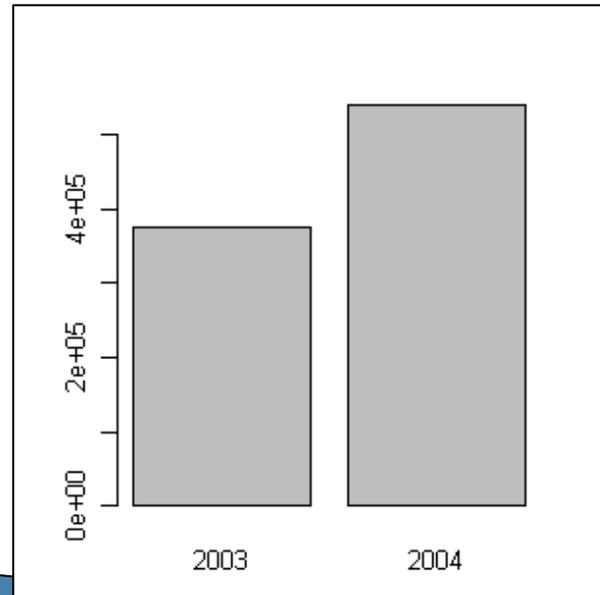
Captures Française pour 2003



Captures Française pour 2004



`barplot(somme[-1])`



**PostgreSQL Unicode ODBC Driver (psqlODBC) Setup**

Data Source:       Description:

Database:       SSL Mode:

Server:       Port:

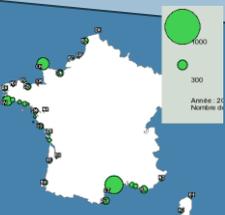
User Name:       Password:

Une table de description des traits (date de l'opération, position, durée, paramètres environnementaux...) et une table des captures spécifiques réalisées pour chaque trait.

```
public_trait
*
code_trait
code_reference
numero_du_trait
date_trait
code_engin
valide
heure_de_filage
heure_de_virage
latitude_filage
longitude_filage
est_ouest_filage
latitude_virage
longitude_virage
est_ouest_virage
latitude_moy
long_moy
est_ouest_moy
duree_du_trait
profondeur
```

```
public_capture_poissons
*
code esp trait
code_trait
code_sp
nbre_ds_ech
pd_ds_ech
```

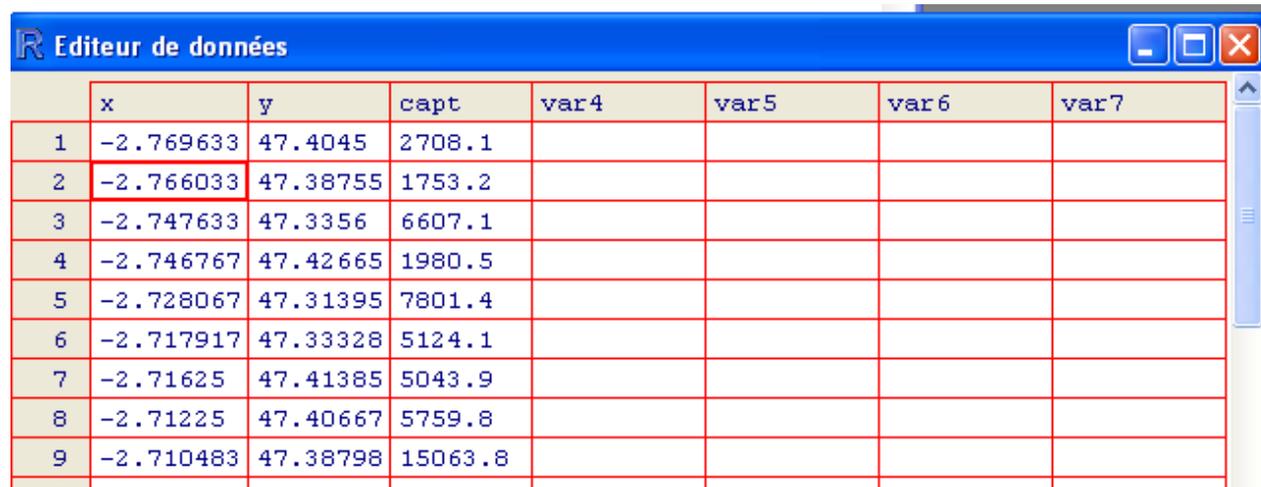
31



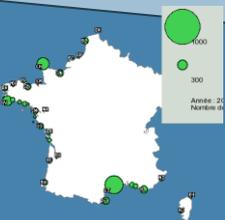
```
select distinct X(pos_filage) as X,Y(pos_filage) as Y,sum(pd_ds_ech) as capt from
trait inner join capture_poissons using(code_trait)
where code_reference=129 group by X,Y
```

J'obtiens les longitudes, latitudes et la somme des captures pour chaque position de la campagne 129 (campagne d'échantillonnage en baie de Vilaine)

**edit(result)**



	x	y	capt	var4	var5	var6	var7
1	-2.769633	47.4045	2708.1				
2	-2.766033	47.38755	1753.2				
3	-2.747633	47.3356	6607.1				
4	-2.746767	47.42665	1980.5				
5	-2.728067	47.31395	7801.4				
6	-2.717917	47.33328	5124.1				
7	-2.71625	47.41385	5043.9				
8	-2.71225	47.40667	5759.8				
9	-2.710483	47.38798	15063.8				



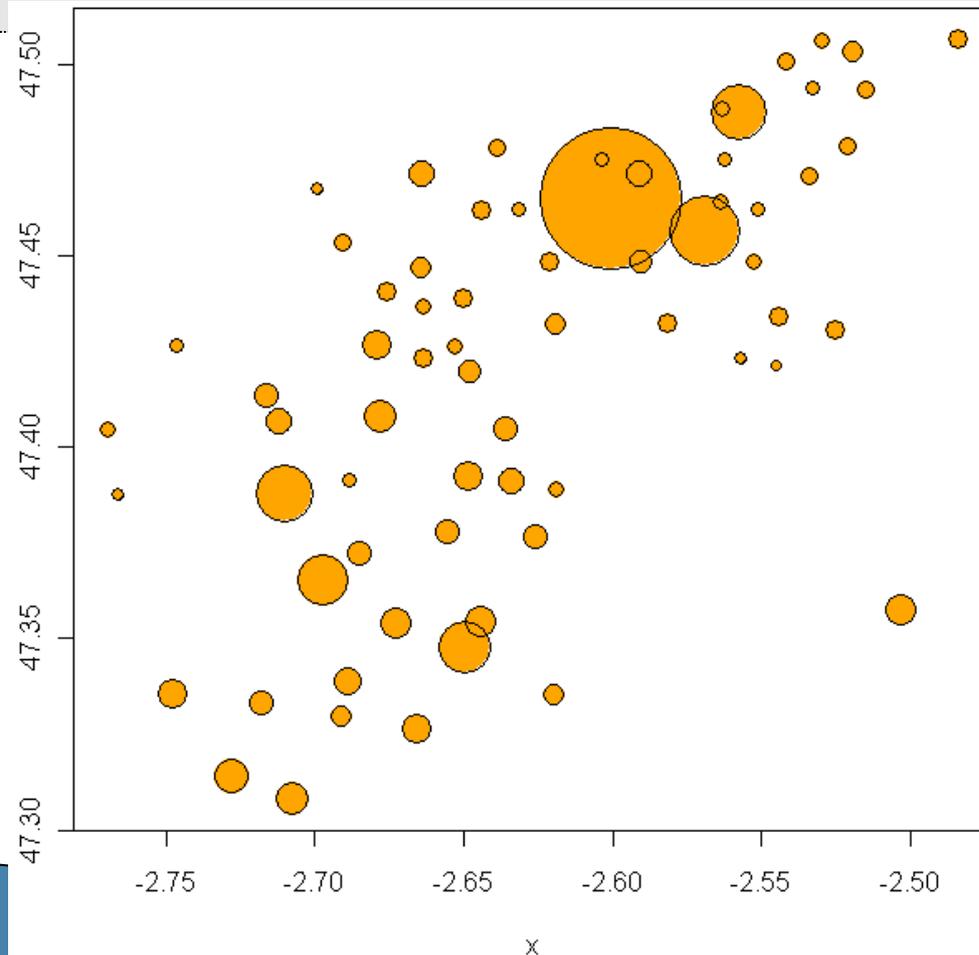
# 8 A vous de jouer

```
max.symbol.size=12
min.symbol.size=1
```

```
plotvar <- 1:10
```

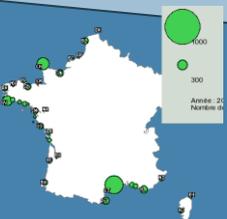
```
symbol.size <- ((plotvar-min(plotvar))/(max(plotvar)-min(plotvar))*(max.symbol.size-min.symbol.size)+min.symbol.size)
```

```
plot(1:10, 1:10, size=symbol.size, col="orange", xlim=c(-2.75, -2.50), ylim=c(47.30, 47.50))
points(1:10, 1:10, size=symbol.size, col="orange", xlim=c(-2.75, -2.50), ylim=c(47.30, 47.50))
```



33

Jérôme Guillon  
Module SIG  
2011-2012



De la cartographie autrement, comment, pourquoi ?

```
densite<-readShapeSpatial("td_sig/nourricerie/stratmoy.shp")
```

```
plotvar<-densite@data$DENS
```

```
nclr <- 6
```

```
plotclr <- brewer.pal(nclr,"Greens")
```

```
class <- classIntervals(plotvar, nclr, style="quantile")
```

```
colcode <- findColours(class, plotclr, digits=0,cutlabels=TRUE)
```

```
colcode
```

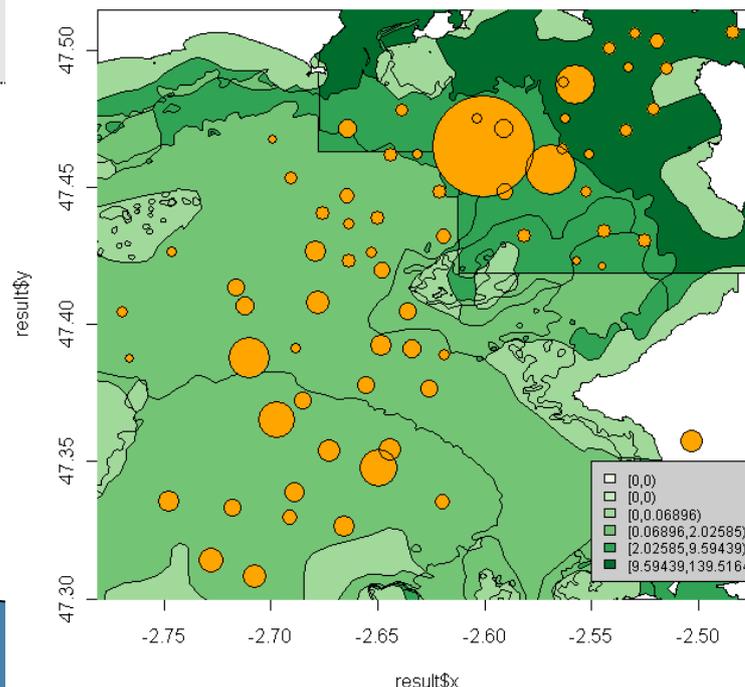
```
plot(result$x,result$y,pch=19, cex=symbol.size,col="orange")
```

```
plot(densite,col=colcode,add=T);
```

```
points(result$x,result$y,pch=19, cex=symbol.size,col="orange")
```

```
points(result$x,result$y,pch=1, cex=symbol.size)
```

```
legend(-2.55, 47.35, legend=names(attr(colcode, "table")), fill=attr(colcode, "palette"), cex=0.75, bty="o",bg='#cccccc')
```



34

Jérôme Guillon  
Module SIG  
2011-2012

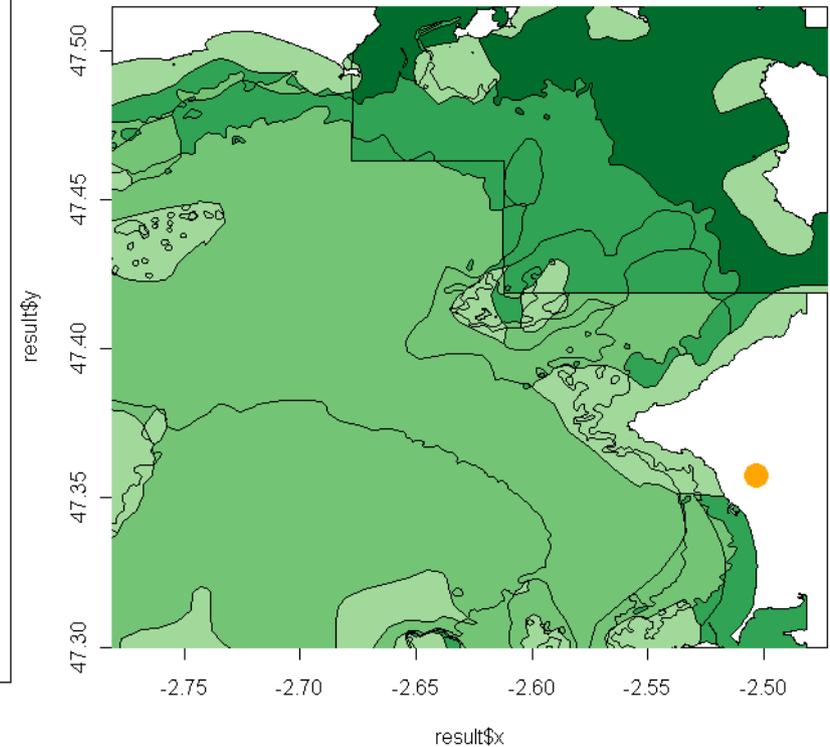
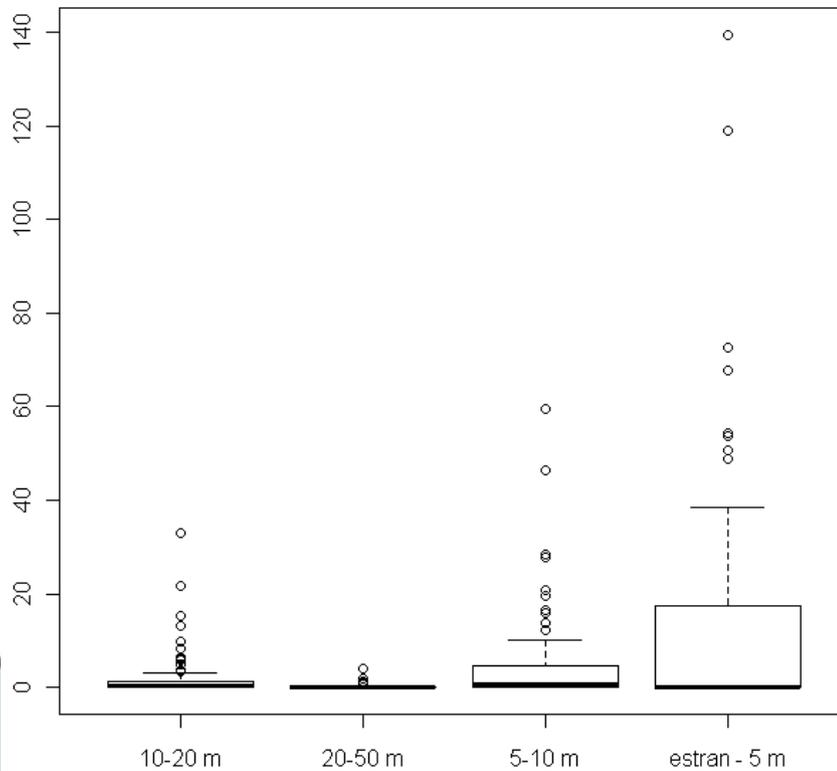


De la cartographie autrement, comment, pourquoi ?

```
plot(densite@data$CLASSES_DE,densite@data$DENS)
```

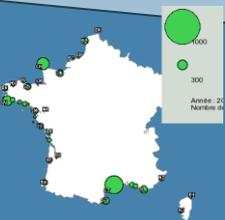
!! Les fichiers shapefiles contiennent des formes mais aussi des informations.

!! Ne pas envisager ces informations qu'en terme de cartographie.



35

Jérôme Guillon  
Module SIG  
2011-2012

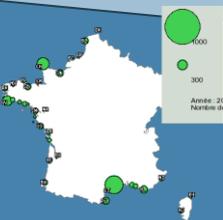


De la cartographie autrement, comment, pourquoi ?

Fin ...

36

*Jérôme Guitton*  
*Module SIG*  
*2011-2012*



De la cartographie autrement, comment, pourquoi ?