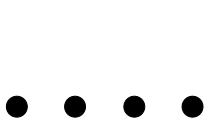
Du point à la surface : interpolation et interaction spatiale

Hadrien Commenges & Timothée Giraud

SEMIN-R, Paris, 10 juin 2016

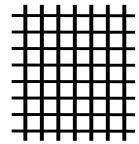
Transformation des objets géographiques



Points

Lignes

Zones



Champs

	Points	Lignes	Zones	Champs
Points	***Barycentre***	***Réseau***	***Buffer*** ***Voronoi***	
Lignes	***Intersection***	***Arbre couvrant***	...	...
Zones	***Centroïde***	...	...	...
Champs	...	...	...	...

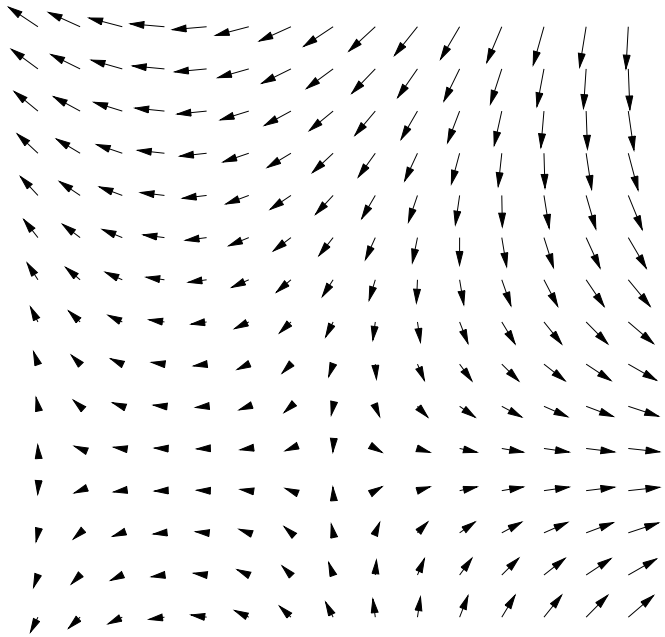
Qu'est-ce qu'un champ ?

CHAMP SCALAIRE

Fonction qui associe un scalaire à chaque point de l'espace de référence

CHAMP VECTORIEL

Fonction qui associe un vecteur à chaque point de l'espace de référence

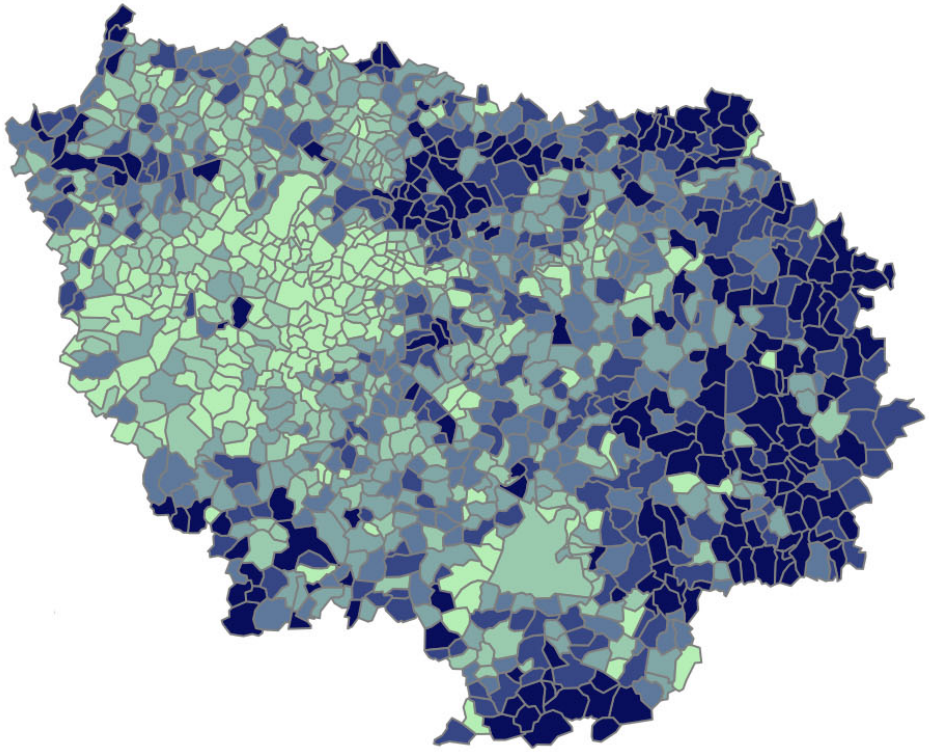


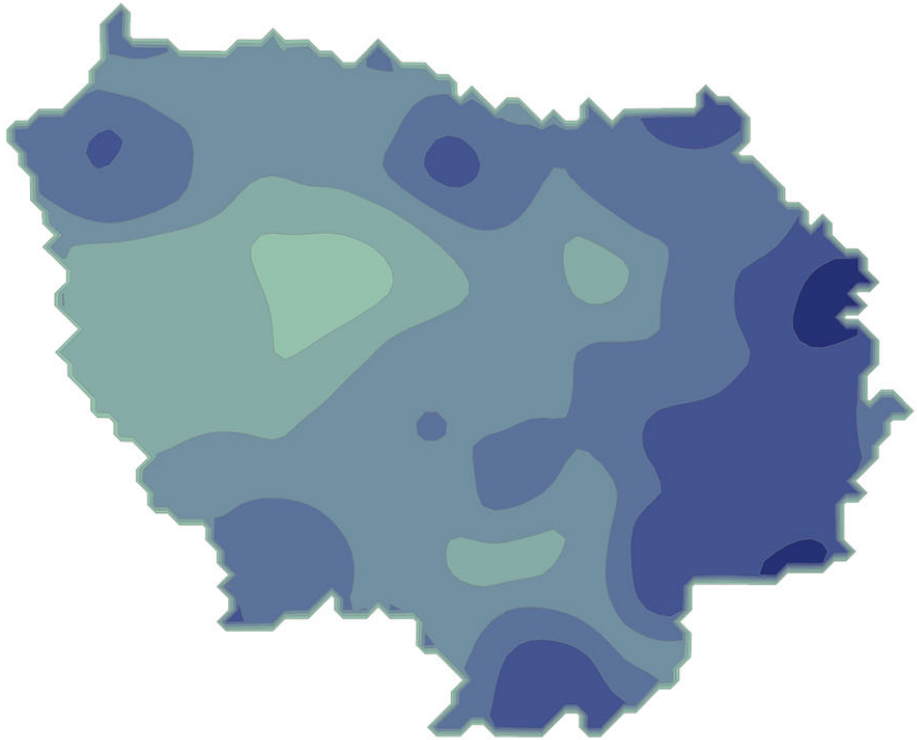
Continuité : à tout point de l'espace de référence correspond une valeur de champ

Unicité : à tout point de l'espace de référence correspond une unique valeur de champ

Quels usages ?

Calcul **vs.** Visualisation





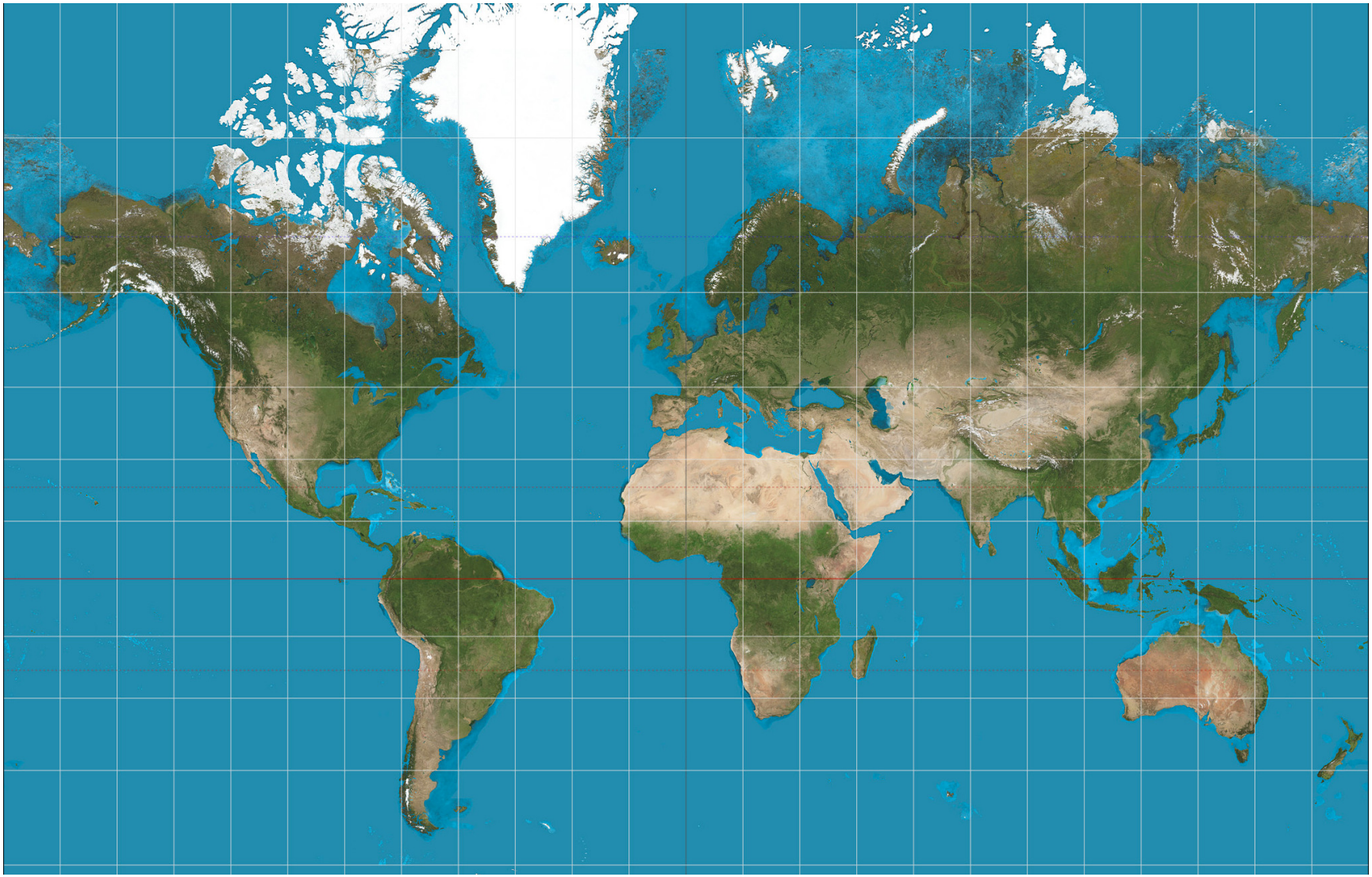
Quelles méthodes ?

	Points 	$f()$	Champ 
DENSITÉ	Non valués	Localisation	Densité de points
INTERPOLATION	Valués (univarié)	Localisation	Valeur prédite qui s'interprète comme la variable initiale (sens et étendue)
INTERACTION SPATIALE	Valués (univarié)	Localisation	Valeur prédite qui s'interprète comme un potentiel d'accès à la variable initiale
RÉGRESSION SPATIALE	Valués (multivarié)	Localisation Régresseurs	Résultat d'un modèle de régression dont les coefficients sont localement pondérés

Quels packages et quelles fonctions ?

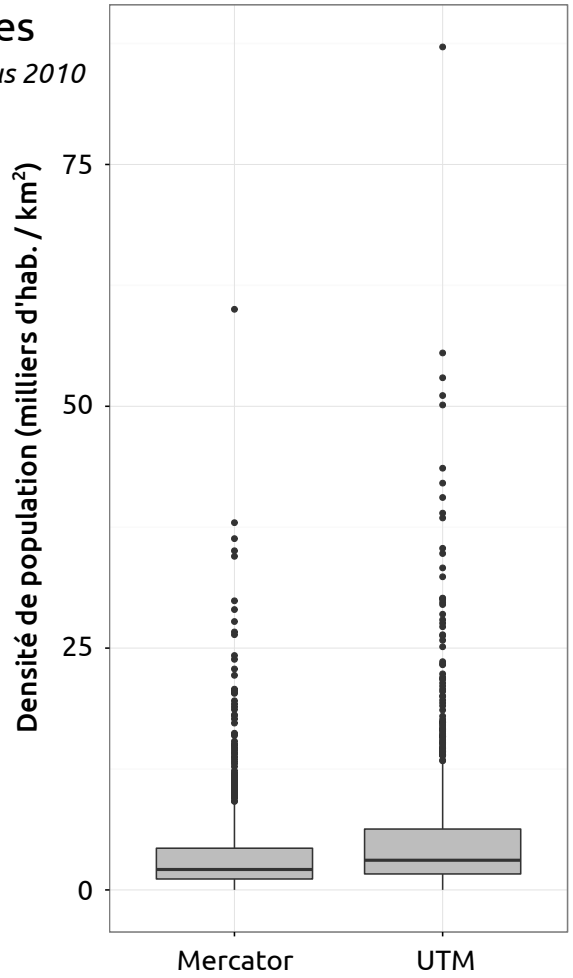
	MÉTHODES	FONCTIONS (PACKAGES)
DENSITÉ	KDE - *Kernel Density Estimator*	`kde2d` (MASS) ; `kernel.factor(spatstat)` ; `geom_density2d` (ggplot2)
INTERPOLATION	Simple : IDW - *Inverse Weighted Distance* Ajustée : *trend analysis*, *kriging*	`idw` (gstat) ; `krige` (gstat) ; `idw` (spatstat) ; `lm` & glm (stats)
INTERACTION SPATIALE	Modèles gravitaires de position (*Stewart's potentials*)	`stewart` ; `reilly` ; `huff` (SpatialPosition)
RÉGRESSION SPATIALE	GWR - *Geographically Weighted Regression*	`gwr` (spgwr)

Quelle projection ?



Source : Wikimedia, Mercator projection

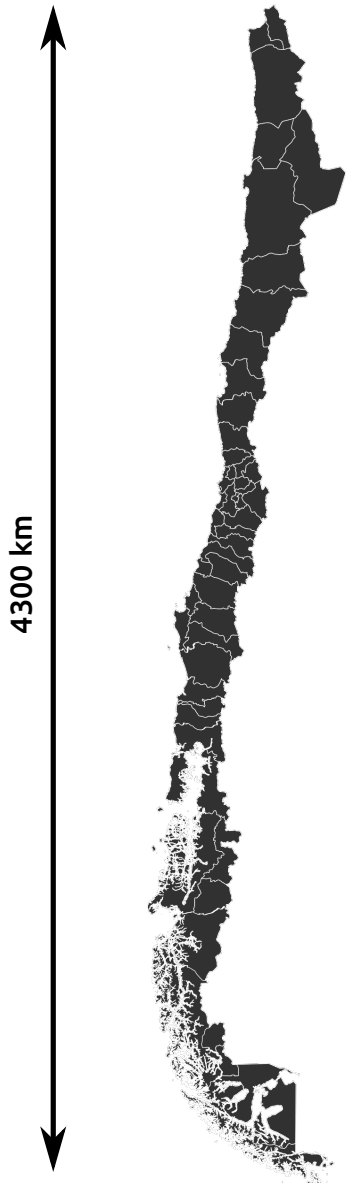
Quelle projection ?



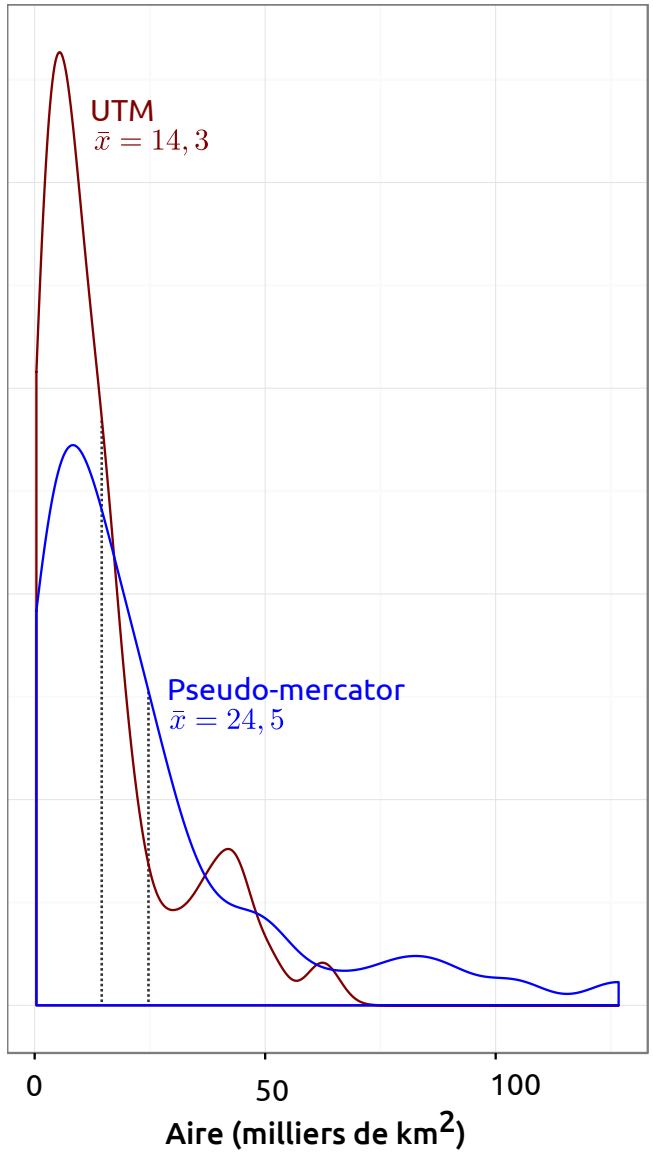
Med. 2111	**Med.** 3069
Moy. 3764	**Moy.** 5483

Paris : 22 000 hab./km²
Île-de-France : 1 000 hab./km²

Quelle projection ?



Aire des provinces chiliennes



Quelle projection ?

Altérations lors de la projection en 2D d'un objet 3D :

- projection conforme : conserve les angles
- projection équivalente : conserve les aires
- calcul de densité $\rightarrow$ ¿altération des aires ?
- **calcul de distances $\rightarrow$ ¿altération linéaire ?**

Quelle distance ?

La distance :

- distance à vol d'oiseau
- distance sur un réseau
- friction : temps, coût

La fonction d'impédance :

- puissance négative ou exponentielle négative
- paramétrage de la fonction (valeur de l'exposant)

Quelle distance ?

- **Méthodes simples** (KDE, IDW) : calibrage de la fonction au jugé
- **Méthodes ajustées** (kriging, gravitaire) : calibrage *data-driven*

Interpolation vs. Interaction

Interpolation (IDW) et interaction s'expriment de façon semblable...

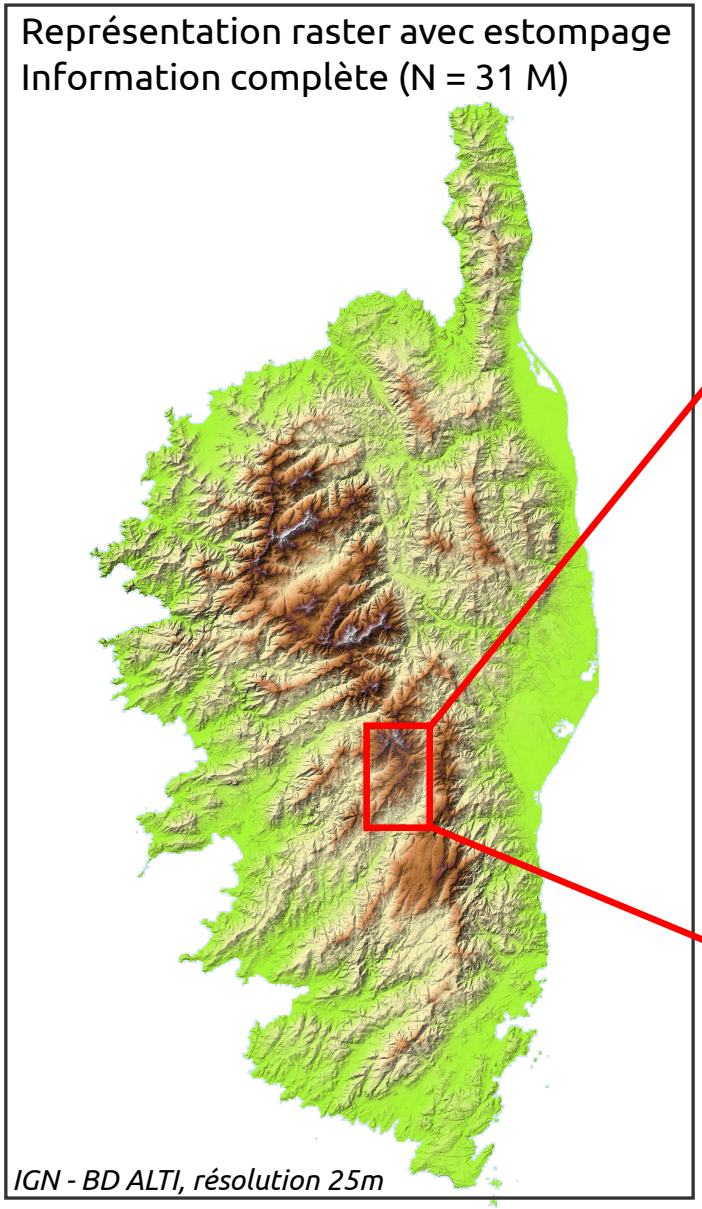
$$z_j^e = \sum_{i=1}^{m} w_{ij} z_i$$

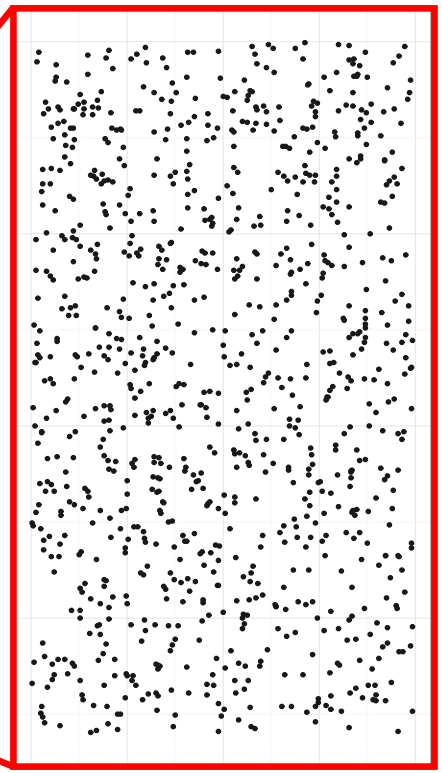
m est le nombre de points de contrôle (qui peut être égal au total n)
w_{ij} est un poids inversement proportionnel à la distance entre i et j

... mais s'appliquent et s'interprètent différemment :

1. Ex. d'interpolation (IDW) de l'altitude en Corse
2. Ex. d'interaction spatiale avec les bars parisiens

Interpolation : l'exemple de l'altitude





Interpolation : l'exemple de l'altitude

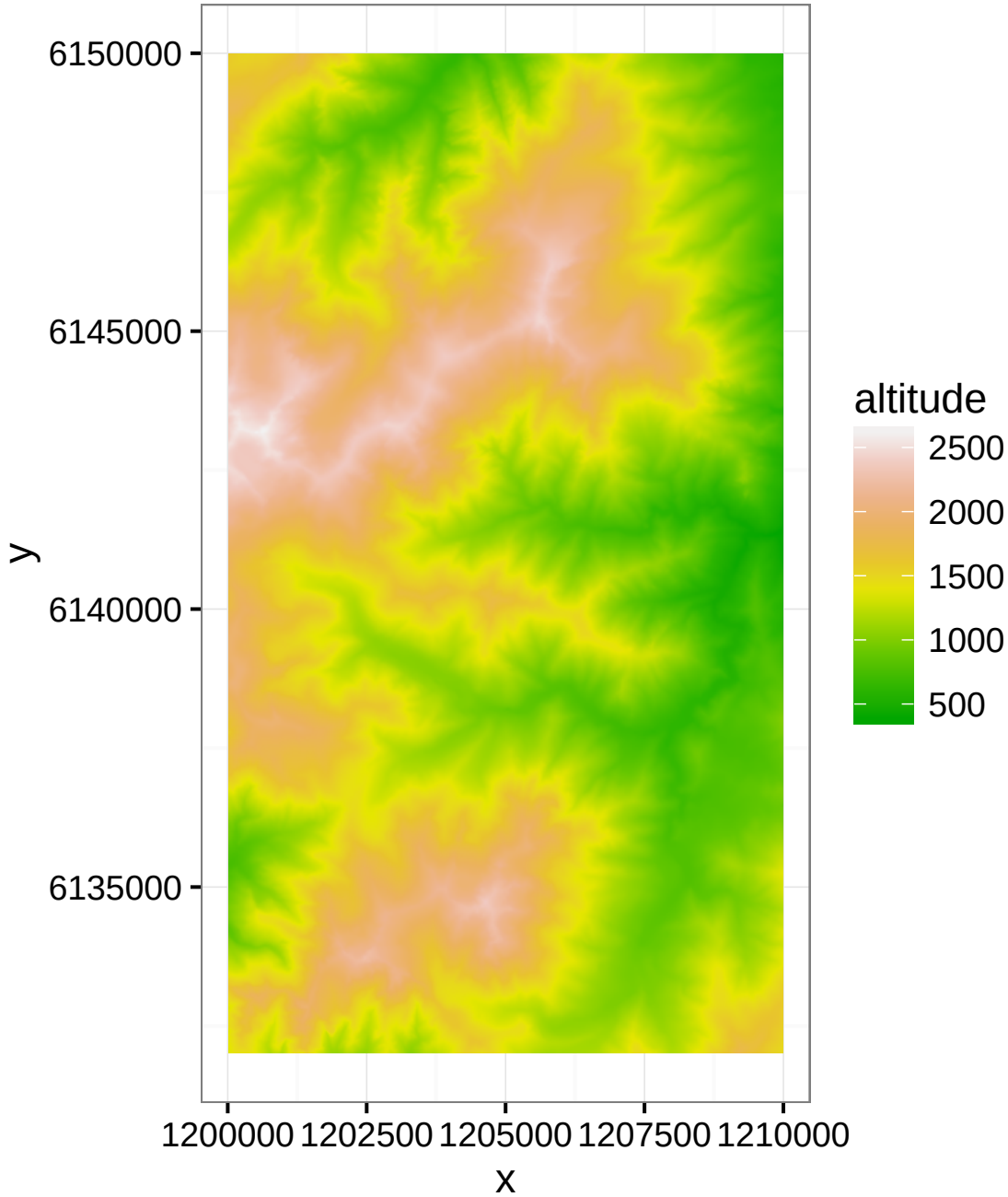
Deux flux de travail possibles (au moins) :

- travail sur objets `data.frame` et affichage avec `ggplot2`
- travail et affichage sur objets `raster` (package `raster`)

Packages nécessaires : `ggplot2`, `sp`, `gstat`

```
str(corseCentre)

## 'data.frame': 573762 obs. of  3 variables:
##  $ x       : num  1200025 1200050 1200075 1200100 1200125 ...
##  $ y       : num  6149975 6149975 6149975 6149975 6149975 ...
##  $ altitude: num  1576 1572 1567 1561 1555 ...

ggplot(corseCentre) +
  geom_raster(aes(x = x, y = y, fill = altitude)) +
  scale_fill_gradientn(colours = terrain.colors(10)) +
  coord_equal() + theme_bw()
```

Interpolation : l'exemple de l'altitude



Interpolation : l'exemple de l'altitude

1. création d'un objet spatial
2. création d'une grille avec un pas fixe (**résolution** du modèle)
3. transformation de la grille en objet spatial

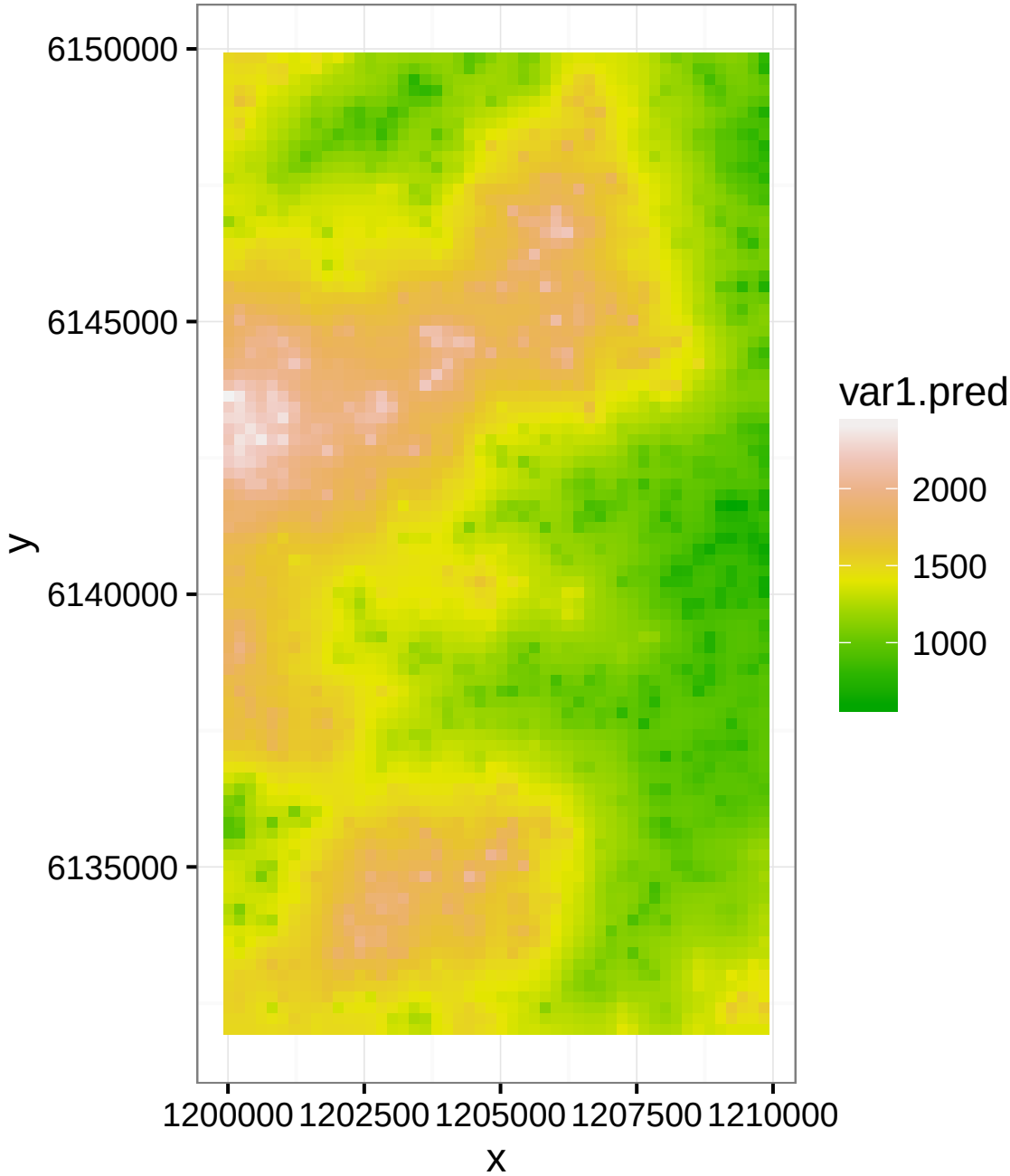
```
# Étape 1
corseSample <- SpatialPointsDataFrame(coords = corseSample[, c("x", "y")],
                                      data = corseSample)
# Étape 2
xRange <- range(coordinates(corseSample)[, 1])
yRange <- range(coordinates(corseSample)[, 2])
fixGrid <- expand.grid(x = seq(xRange[1], xRange[2], by = 200),
                       y = seq(yRange[1], yRange[2], by = 200))

# Étape 3 (identique à l'étape 1)
coordinates(fixGrid) <- ~ x + y
gridded(fixGrid) <- TRUE
```

Interpolation : l'exemple de l'altitude

- calcul du modèle IDW dans la grille
- affichage du résultat (transformé en `data.frame`) avec `ggplot2`

```
corseIdw <- idw(formula = altitude ~ 1,
                locations = corseSample,
                newdata = fixGrid)

## [inverse distance weighted interpolation]

class(corseIdw)

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

ggplot(as.data.frame(corseIdw)) +
  geom_raster(aes(x = x, y = y, fill = var1.pred)) +
  scale_fill_gradientn(colours = terrain.colors(10)) +
  coord_equal() + theme_bw()
```

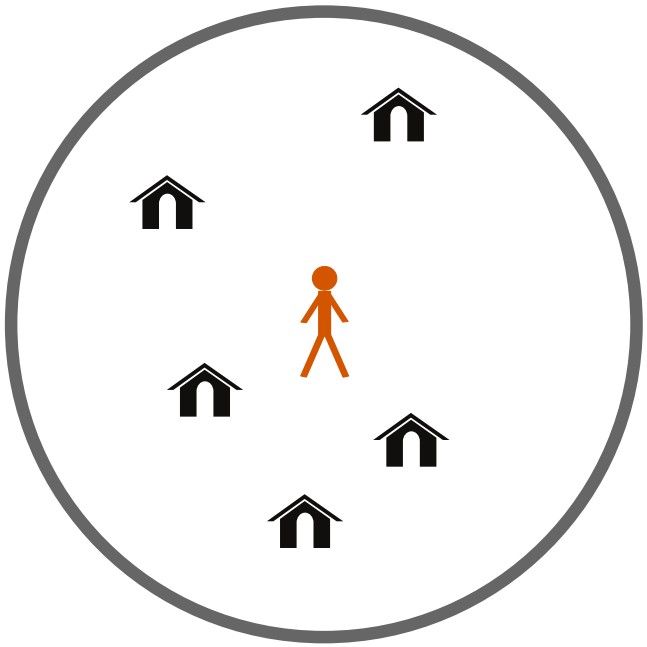
Interpolation : l'exemple de l'altitude

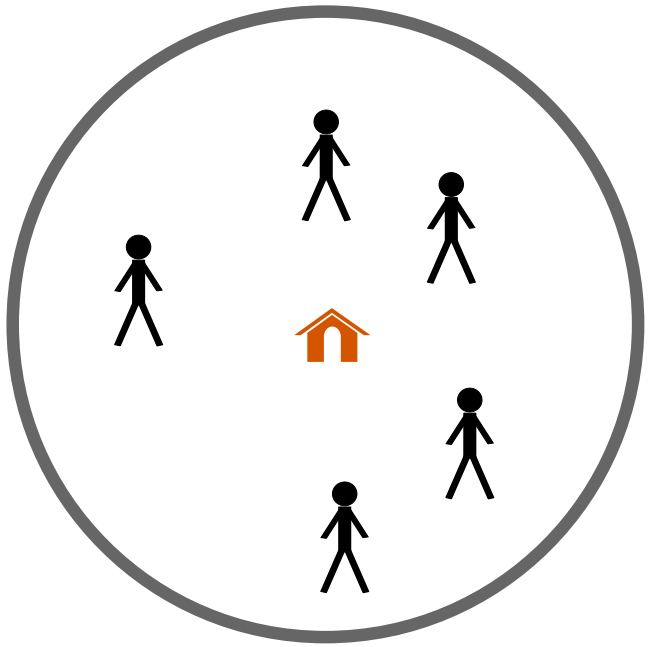


Interaction : l'exemple des bars parisiens

Agents et ressources :

- **Consommateurs** (agents) cherchent **magasins** (ressources)
- **Magasins** (agents) cherchent **clients** (ressources)





Interaction : l'exemple des bars parisiens

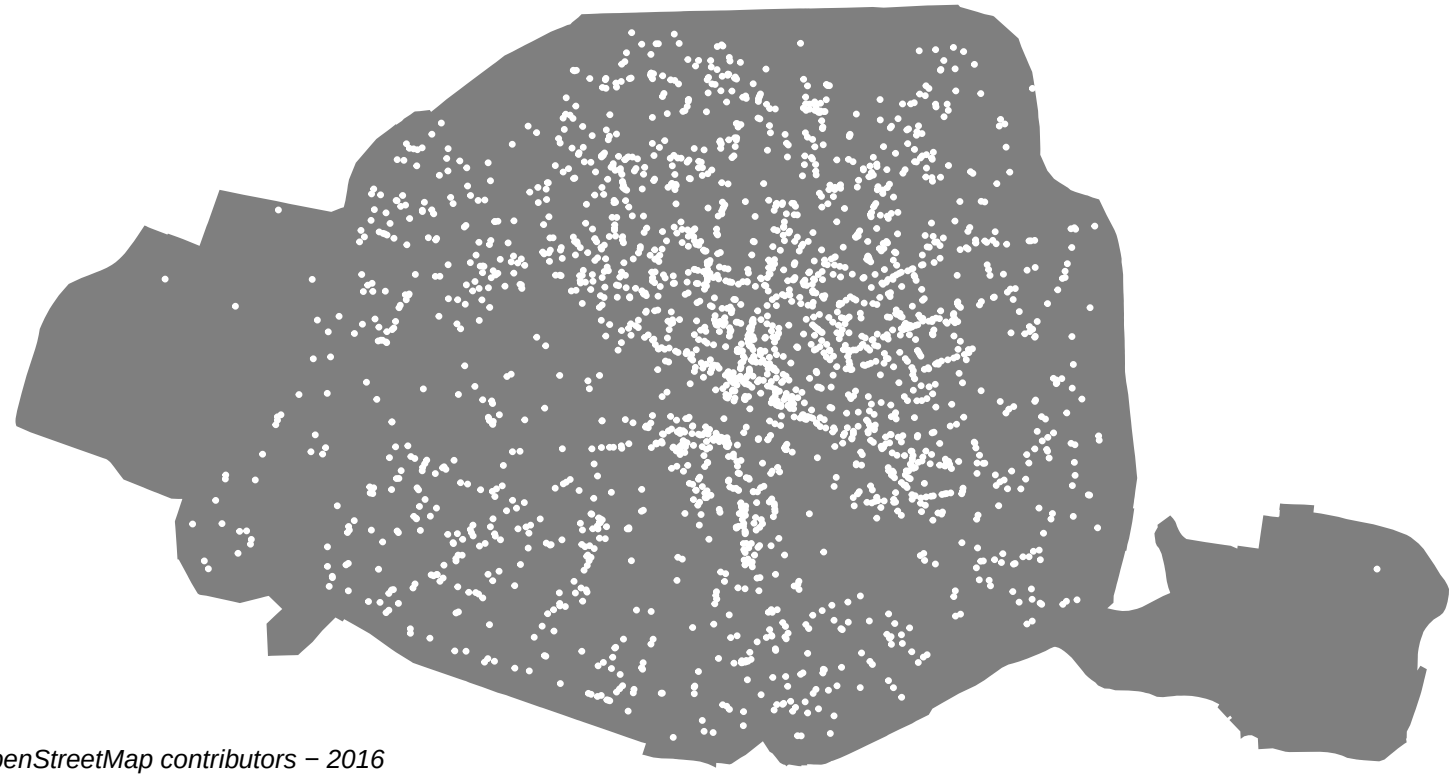
Packages nécessaires :

- sp : formats spatiaux
- cartography : cartographie statistique
- SpatialPosition : interaction spatiale

```
plot(boundParis, col = "grey50", border = NA)
plot(ptBars, pch = 20, col = "white", cex = 0.3, add = TRUE)
layoutLayer(title = "Les bars parisiens",
            frame = FALSE,
            coltitle = "black",
            col = NA,
            scale = NULL,
            sources = "OpenStreetMap contributors - 2016",
            author = "T. Giraud")
```

Interaction : l'exemple des bars parisiens

Les bars parisiens



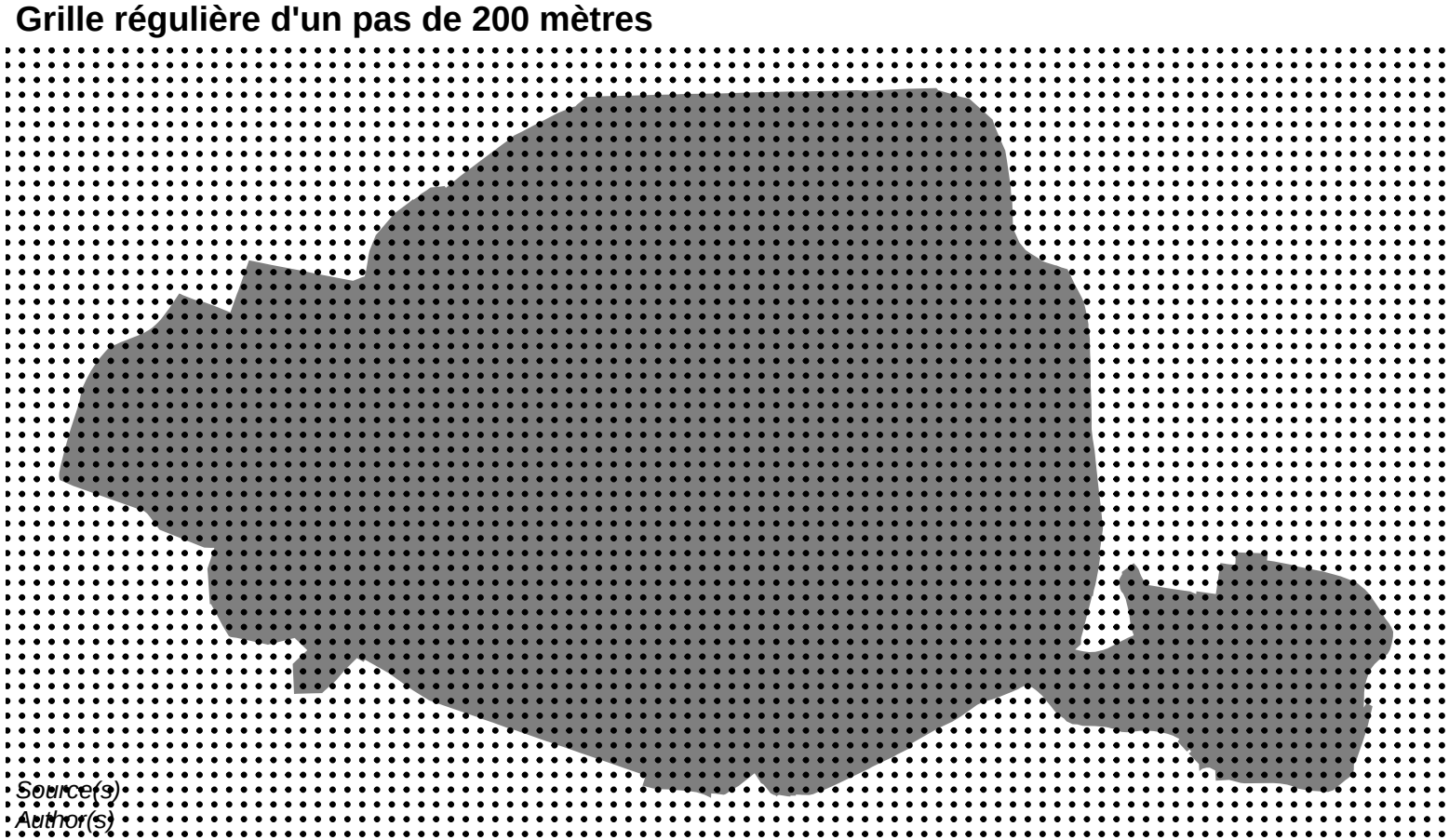
OpenStreetMap contributors − 2016
T. Giraud

Interaction : l'exemple des bars parisiens

Comme pour l'interpolation, le **champ** prend la forme d'une **grille** :

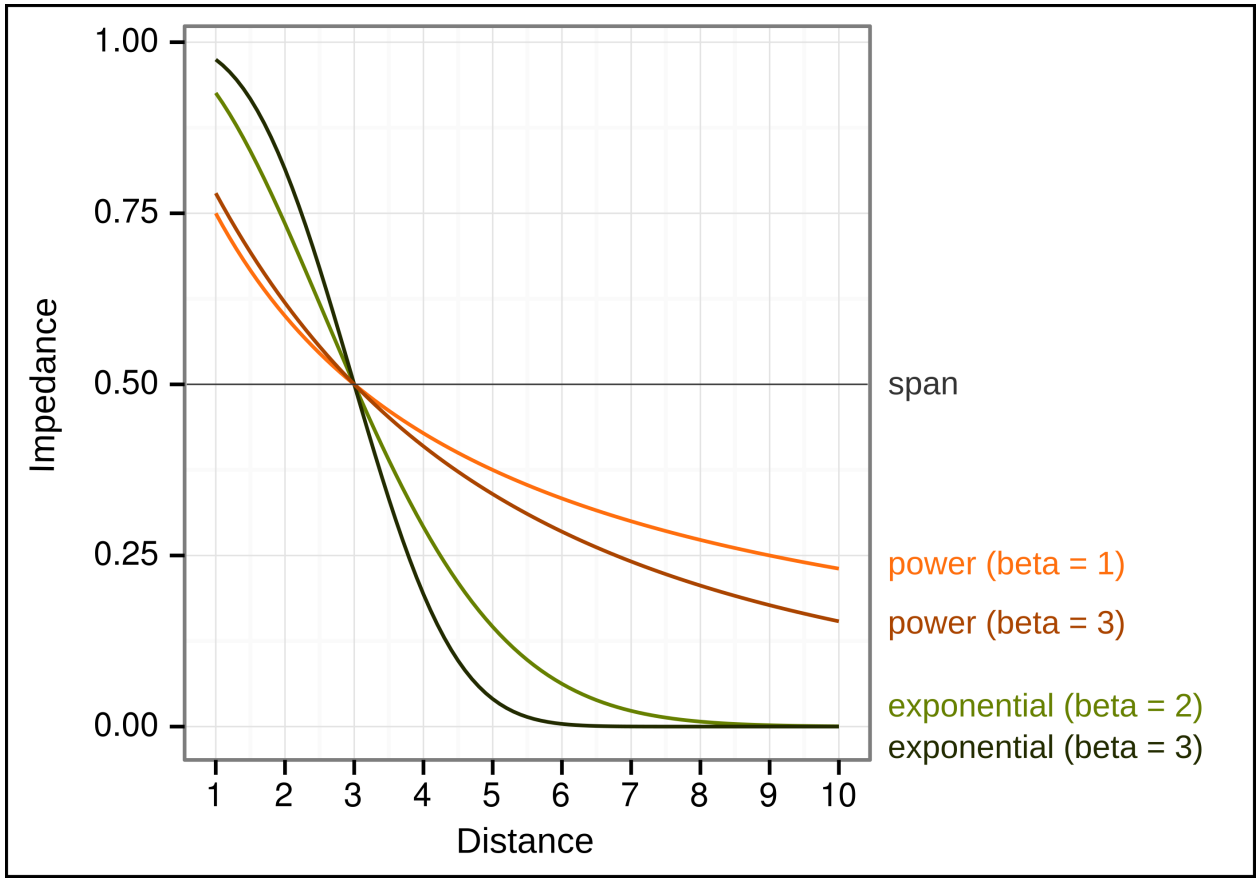
```
# Création de la grille (résolution dans l'unité du syst. de proj.)
gridParis <- CreateGrid(w = boundParis, resolution = 200)

# Affichage de la grille
plot(boundParis, col = "grey50", border = NA)
plot(gridParis, pch = 20, col = "black", cex = 0.3, add = TRUE)
layoutLayer(title = "Grille régulière d'un pas de 200 mètres",
            frame = FALSE,
            coltitle = "black",
            col = NA,
            scale = NULL)
```

Interaction : l'exemple des bars parisiens



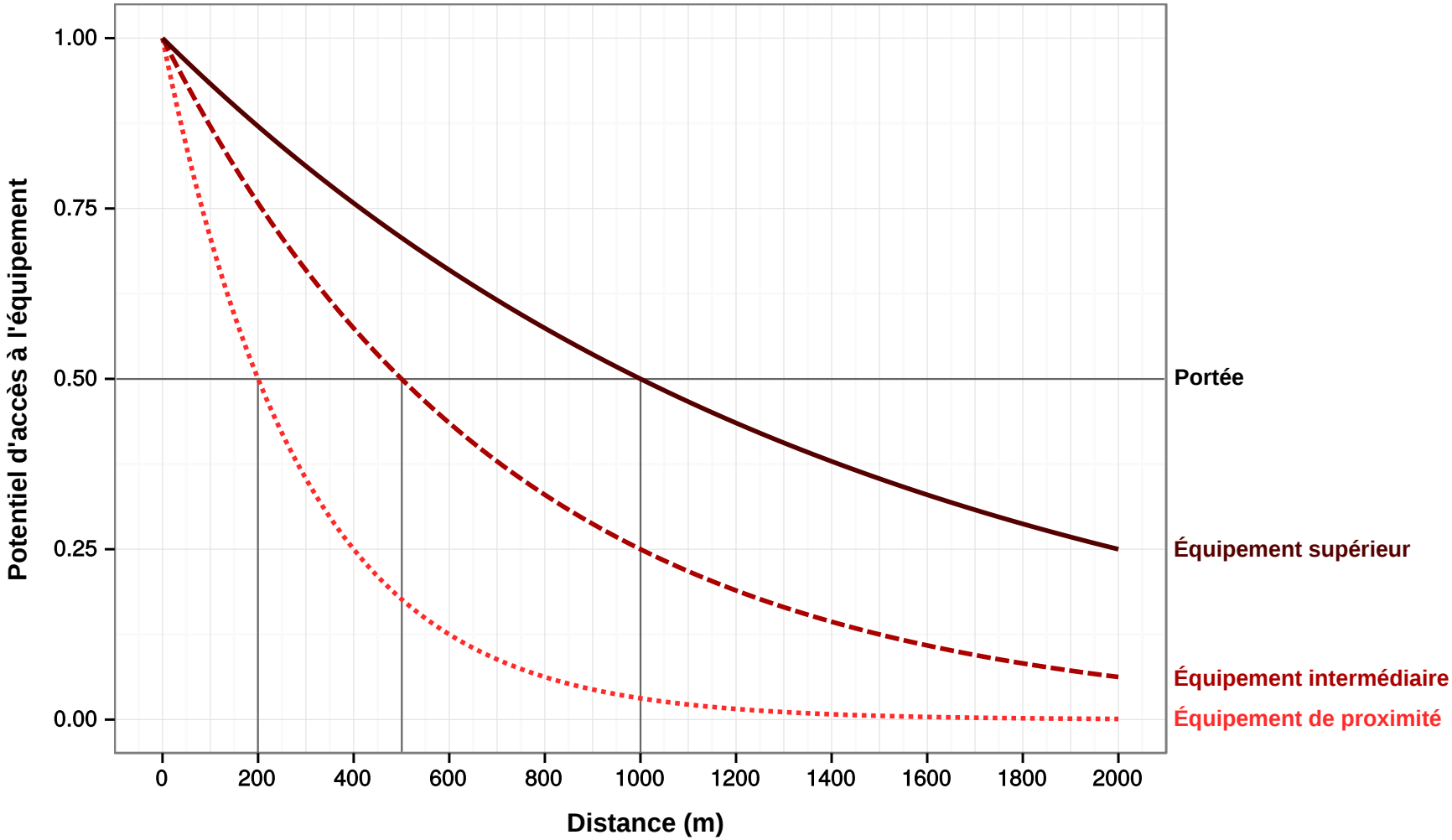
Interaction : l'exemple des bars parisiens

- Puissance négative ou exponentielle négative
- Paramétrage de la fonction (valeur de l'exposant)



Interaction : l'exemple des bars parisiens

Exemple de calibrage :



Interaction : l'exemple des bars parisiens

- Calcul des potentiels sur la grille **incluant calcul de distance**
- Transformation des points en raster
- Transformation du raster en isolignes (contour)

```
potBars <- stewart(knownpts = ptBars,
                   unknownpts = gridParis,
                   varname = "n",
                   typefct = "exponential",
                   span = 1000,
                   beta = 3)

rastPotBars <- rasterStewart(potBars)

contPotBars <- rasterToContourPoly(r = rastPotBars,
                                   mask = boundParis,
                                   breaks = seq(0, 400, by = 50))
```

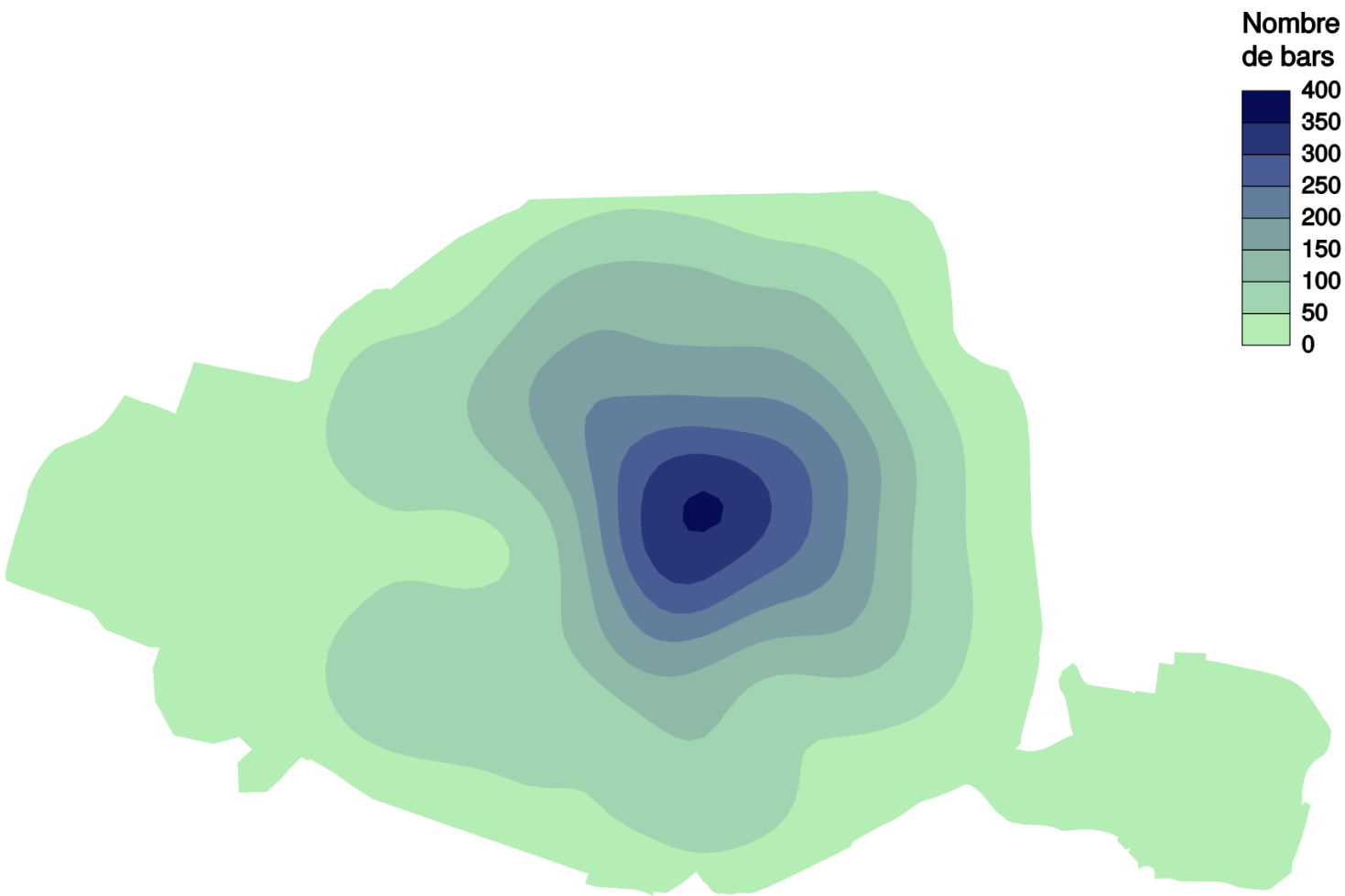
Interaction : l'exemple des bars parisiens

Cartographie des potentiels :

```
choroLayer(spdf = contPotBars,
           df = contPotBars@data,
           var = "center",
           breaks = seq(0, 400, by = 50),
           border = NA,
           legend.title.cex = 0.7,
           legend.frame = FALSE,
           legend.pos = "topright",
           legend.title.txt = "Nombre\nde bars",
           col = carto.pal(pal1 = 'turquoise.pal', n1 = 8),
           add = FALSE)

layoutLayer(title = "Nombre potentiel de bars (voisinage de 1000 mètres)",
            frame = FALSE,
            coltitle = "black",
            col = NA, scale = NULL, sources="",
            author = "")
```

Interaction : l'exemple des bars parisiens

Nombre potentiel de bars (voisinage de 1000 mètres)



Interaction : l'exemple des bars parisiens

- Calcul préalable d'une matrice de distance
- Approximation de la distance réseau (CERTU 2005, coef. 1,3)
- Transformation en temps à une vitesse piétonne (4,5 km/h)
- Bien sûr il est possible de faire un véritable calcul du temps réseau

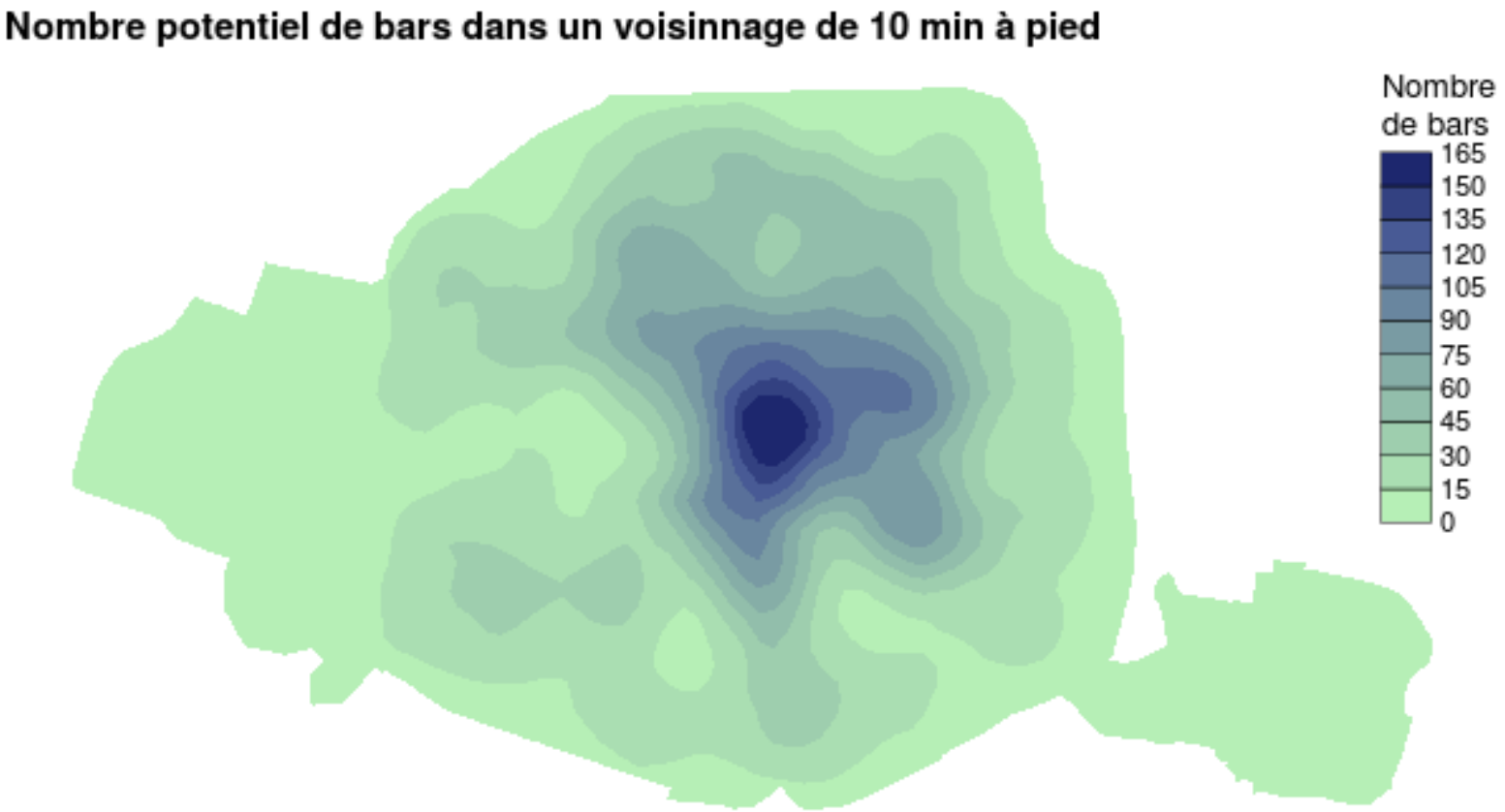
```
# création d'une matrice de distance
distBars <- spDists(x = coordinates(ptBars),
                    y = coordinates(gridParis),
                    longlat = FALSE,
                    diagonal = FALSE)

# approximation d'une distance réseau
distBarsNetwork <- distBars * 1.3

# matrice de temps à pied en minutes
timeBars <- (distBarsNetwork / 1000) / (4.5 * 60)
```

Interaction : l'exemple des bars parisiens

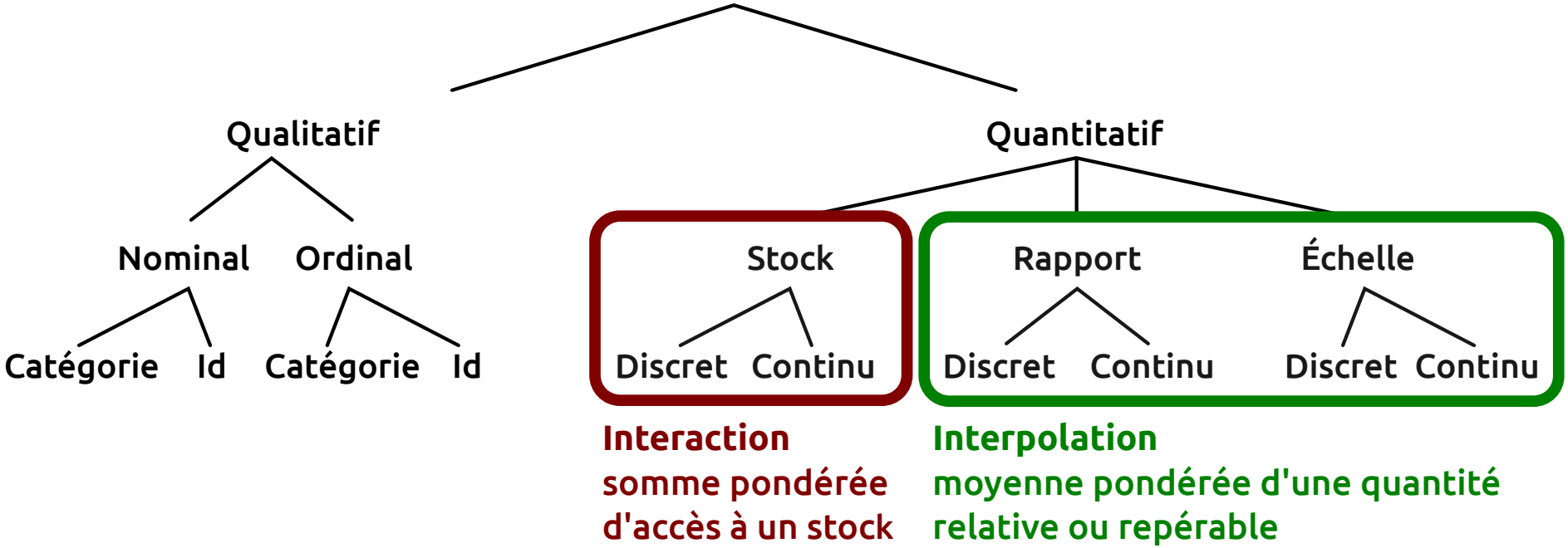
- Calcul des potentiels sur la grille **avec calcul préalable d'une matrice de distance**
- Transformation des points en raster
- Transformation du raster en isolignes (contour)

```
potBars <- stewart(knownpts = ptBars,
                   unknownpts = gridParis,
                   varname = "n",
                   matdist = timeBars,
                   typefct = "exponential",
                   span = 10,
                   beta = 3)

rastPotBars <- rasterStewart(potBars)

contPotBars <- rasterToContourPoly(r = rastPotBars,
                                   mask = paris,
                                   breaks = seq(0, 170, by = 15))
```

Interaction : l'exemple des bars parisiens



Conclusion



La valeur connue d'un point échantillonné est-elle la vraie valeur que l'on cherche à estimer ?