INTRODUCTION AUX MÉTHODES DE GÉOSTATISTIQUE

SUPPORT DE FORMATION

LIFFRÉ (35), DÉCEMBRE 2019

Intervenant : Loïc GAUDIN
Docteur de l'Université de Rennes1
loic.gaudin@arkeomap.com

Introduction

Ce document constitue un support aux concepts de base de la géostatistique. Il contient des explications et surtout des exemples d'applications.

Les objectifs sont de découvrir les différents concepts de base de la géo-statistique à savoir principalement :

- les calcus de moyennes, écart-types, variances, covariances, coefficients de corrélation, régressions linéaires, (principe des régressions polynomiales), analyses multivariées et quelques tests statistiques,
- les principales méthodes d'interpolations : Splines, Pondération par l'Inverse de la Distance (IDW) et Krigeage.

L'objectif de cette formation étant davantage de fournir un bagage méthodologique que théorique (“mathématiques purs”, démonstrations), ce support permet d'illuster le propos par des exemples de calculs concrets.

Pour ces exemples nous utiliserons un logiciel libre : R accompagné de l'interface Rstudio et des librairies Gstat et GeoR...

Le logiciel Qgis sera utilisé pour les applications nécessitant la manipulation des données spatiales.

Table des matières

Chapitre 1 : Installation de R

R est un logiciel libre de traitement des données et d'analyses statistiques mettant en œuvre le language de programmation S. C'est un projet qui a été fondé sur l'environnement développé dans les laboratoires Bell.

C'est un logiciel libre distribué selon les termes de la licence GNU GPL (Licence Publique Générale de distribution des Logiciels Libres) et est disponible sous GNU/Linux, FreeBSD, NetBSD, OpenBSD, Mac OS X et Window.

Si R dispose dans sa version de base de la plupart des fonctionnalités utiles pour la statistique courante, ses possibilités s'élargissent dès que l'on utilise les paquets (ou « extensions »), souvent écrits en R et mis librement à disposition.

La documentation, le téléchargement de R ainsi que différents paquets peuvent être effectués depuis le site dédié :

http://www.r-project.org/

L'utilisation de R peut être facilitée par l'utilisation des paquets suivants :

- Rstudio: offre une interface graphique (sinon il faut travailler dans une console (http://www.rstudio.com)
- gstat: permet de réaliser les interpolations de type krigeage (avec variogrammes) et IDW. (http://cran.r-project.org/web/packages/gstat/index.html

- GeoR : permet le krigeage (http://leg.ufpr.br/geoR/geoRdoc/geoRintro.html#interpolation

- Les interpolations des Splines (function “splinefun”) sont incluses dans les dernières versions de R.

1.1 Installation de R sur Windows

Télécharger l'executable depuis le site http://www.r-project.org/ ou http://cran.univ-lyon1.fr/

Il existe une FAQ dédiée à l'installation de R pour Windows. (http://cran.univ-lyon1.fr/).

Prendre la version de base. (R.3.6.0 ou plus récente for Windows).

Aller sur le site du CRAN le plus proche et choisir une version de R pour windows (extension ".exe" et non "tar.gz" réservé à Linux).

Une fois l'executable téléchargé, choisissez la version 64 ou 32 bits en fonction des caractéristiques de votre ordinateur puis installez le logiciel dans C:programmes.

Il est possible de télécharger une interface graphique Rstudio fonctionnant pour Windows (http://www.rstudio.com/products/rstudio/download/). L'interface propose de nombreuses fonctionnalités intéressantes, telles la coloration syntaxique, la gestion des fichiers Sweave et LaTeX, un tableur, un gestionnaire de librairies, etc.

Installer la dernière version de Rstudio (Rstudio-1.2.1335) ou plus récente for Windows, sur "C:\Program Files\RStudio".

1.2 Installation de R sur Linux (UBUNTU)

L'installation de **R** est très simple : il suffit d'installer le paquet **r-base**. R dispose de nombreuses fonctions supplémentaires disponibles sous la forme de paquets téléchargeables. Pour pouvoir installer certains de ces paquets, il vous faut de quoi les compiler. C'est pourquoi il est aussi conseillé d'installer le paquet **r-base-dev**.

En ligne de commande :

sudo apt-get install r-base

R est alors installé sur votre ordinateur mais vous ne voyez rien car il n'y a pas d'interface graphique. On peut le lancer depuis le terminal avec la commande R. Il existe plusieurs interfaces graphiques, dont RKward et Rcmdr, et Rstudio que nous utiliserons.

RStudio est un environnement de développement intégré. RStudio propose de nombreuses fonctionnalités intéressantes, telles la coloration syntaxique, la gestion des fichiers Sweave et LaTeX, un tableur, un gestionnaire de librairies, etc. Pour l'installer, il suffit de télécharger le fichier .deb sur le site de Rstudio (http://www.rstudio.com/products/rstudio/download/) et de l'ouvrir à l'aide de la logithèque Ubuntu (option par défaut).

Pour connaitre la version du gestionnaire (32 ou 64 bit)

~$ uname -mrs

Puis choisir la version téléchargeable sur le site de Rstudio:

ex. RStudio 0.98.1028 - Debian 6+/Ubuntu 10.04+ (64-bit)

Télécharger le fichier sur sa machine et le lancer à partir du gestionnaire de fichiers (veiller à ce que le gestionnaire soit bien à jour : ex. Err. “Unknown media type in type 'all/all'” : a nécessité de mettre à jour KDE :
sudo rm /usr/share/mime/packages/kde.xml
sudo update-mime-database /usr/share/mime
sudo dpkg -i rstudio-0.98.1028-amd64.deb)

Rstudio est installé dans Racine/usr/lib/rstudio/bin

1.3 Interface Rstudio

– **pour installer des packages : onglet “tools” → install package**

L'interface propose directement les packages proposés par le CRAN : *The Comprehensive R Archive Network.* Dans ce cas c'est la dernière version du package qui est prise. Or il peut y avoir des incompatibilités avec la version de R (ex. Dernière Version de spacetime).

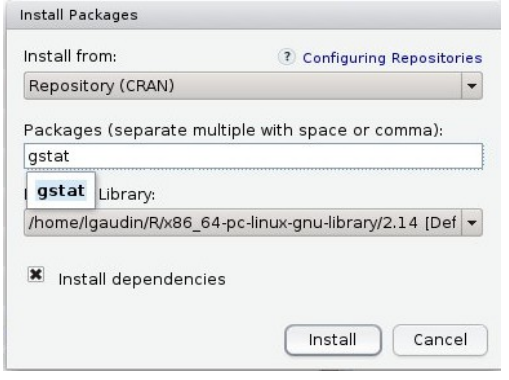
Ou bien il faut télécharger la librairie tar.gz et pointer dessus.

– **Exemple installation de Gstat : pour la version R.3.2.4 de R :**

- installer xts (via le CRAN) avec les dépendances

- tenter l'installation de gstat (via le CRAN). Si nécessaire, télécharger le package spacetime 1.0.1.tar.gz puis installer le package.

- installer gstat (via le CRAN) avec les dépendances



– **Exemple installation de GeoR : pour la version 2.14 de R :**

- tenter l'installation de GeoR (via le CRAN). Si nécessaire : installer la version RandomField 2.0.71.tar.gz

- installer GeoR (via le CRAN)

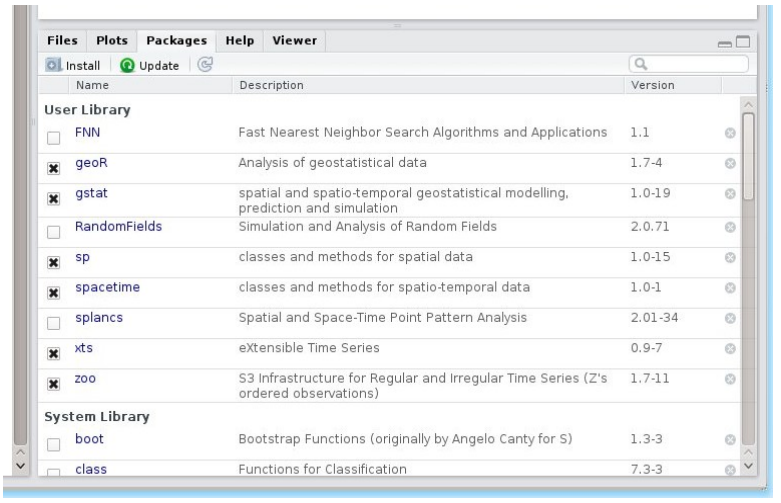
– **pour voir si les packages sont installés : onglet “view” / “show package”**

– **Installer** : akima, (pour interp), gstat, geoR, Raster, sp, spaceTime, xts, zoo, R6, FactoMineR

RQ. Pour FactoMineR : bien installer R6, au besoin l'installer avant

– **pour les charger (les activer) cocher les cases correspondantes**

– Si nécessaire, les installer via le CRAN.



Exemple d'interface de parametrage des packages

1.4 Lancement en ligne de commande

Utilisez l'espace “console” (espace en bas à gauche de Rstudio) pour saisir des lignes de commande.

Ecrivez les commandes suite au prompt : “>”

Pour avoir des détails sur une fonction : taper dans la console “?” suivi de la fonction. Exemple : `> ?interp`

RQ. Vérifier que les librairies nécessaires sont bien chargées sur le serveur.

Exemple : dans la console

RQ. Saisir "help(--x--)" ou ?--x-- pour avoir de l'aide sur une commande,

RQ. Saisir print(--objet--), pour avoir une description de l'objet

RQ. Pour charger un fichier avec un scripts : aller dans l'onglet Files puis sélectionner le fichier txt contenant le script souhaité (ex. scripts_1_moyenne_ec_var.txt).

RQ. Pour définir le répertoire courant comme répertoire de travail (*More → Set Working Directory*). Au besoin, aller dans *Session → Set Working Directory → Choose Directory* pour changer le “répertoire Racine”.

RQ. Il est possible d'importer des données ou tableaux de données. Par exemple, le chargement du fichier `dataAFC.csv` (situé dans /Donnees/Donnees_Chapitre_2) se fait à l’aide de la commande `read.csv2()`, qui permet de lire des fichiers CSV pour lesquels le séparateur de champ est une tabulation.

Les données seront par exemple associées à la variable `dataAFC` à l’aide de l’opérateur d’affectation `<-`.

Dans ce qui suit, on supposera que l’utilisateur a bien défini le répertoire contenant le fichier `afc_tache_menageres.txt` comme répertoire de travail, soit à l’aide de la commande `setwd()` soit via le menu de Rstudio.

```
DataAFC<-
read.csv2(file="afc_tache_menageres.txt",sep="\t",
header=TRUE, row.names=1, check.names=FALSE);
```

RQ. On peut aussi indiquer le chemin du repertoire à partir de l'endroit où a été fixé la racine. Exemple :

```
dataAFC<-
read.csv2(file="../Donnees/Donnees_Chapitre_2/afc_tach
e_menageres.txt",sep="\t",  header=TRUE,  row.names=1,
check.names=FALSE);
```

Saisissez `>print(dataAFC)` pour visualiser l'objet dans la console.

Pour pouvoir executer les commandes R en ligne de commande, il faut enregistrer l'ensemble des commandes dans un fichier.txt (fait avec R studio par exemple), puis charger ce fichier depuis la console (ou exec en PHP).

RQ. possibilité aussi de faire des fichiers."sh", permettant d'appeler des executions "php" puis "R".

R --no-save -f chemin/fichier.txt

"R" pour lancer une commande "R"
"--no-save" pour ne pas sauvegarder les variables
"- f " : pour l'execution des commandes du fichier.txt

Exemple de script .sh

```
#!/bin/bash


LOG="deroulement/deroulement_${1}"
[ -f $LOG ] && rm $LOG && touch $LOG
PATTERN="Messages d'avis"
PATTERN2="Warning messages:"

{2}

echomsg()
{
msg="$1"
echo -en "\033[32m";echo -e "\n\n################################\n$msg\n################################\n"
echo -en "\033[32m";echo -e "\n\n################################\n$msg\n################################\n" 2>&1 >> $LOG
tput sgr0
}


# $1 et $2 correspondent à des informations passées en paramètre lors de l'appel du fichier.sh (executable)
NUM_DEPT=${1}
DIR_HOME=$

# definitions de variables et de "pointeurs" sur fichiers, à noter le fichier R_FILE contenant un script "R"
# repertoire courant :
HOME_DIR="/home/ariase/Bureau/Automatisation-new"
NUM_DEPT=${1}
LOG_FILE="${HOME_DIR}/log/log_${NUM_DEPT}.txt"
SCRIPT_DIR="${HOME_DIR}/script"
#fichier contenant le script R
R_FILE="${HOME_DIR}/temp/R_FILE.txt"
#fichier générateur de couleurs
C_FILE="${HOME_DIR}/temp/C_FILE.sh"
DEPT_SRC="${HOME_DIR}/dept/dep_${NUM_DEPT}.csv"
GRILLE_SRC="${HOME_DIR}/grille/dep_${NUM_DEPT}.csv.txt"
IMAGE_DIR="${HOME_DIR}/images"
KML_DIR="${HOME_DIR}/KML"
ASCII_DIR="${HOME_DIR}/exportAscii"

dateDebut=`date '+%y/%m/ùd a %H:%M:%s'`
echo "${NUM_DEPT} débuté le ${dateDebut}" >> ${LOG_FILE}

echomsg "writeRfilev1"


# chargement et lancement d'un executable php avec en paramètre différentes variables et pointeurs définis précédemment. ????
# le but ici est d'écrire du script "R" de façon dynbamique au moyen d'un fichier php (dans ${R_FILE})
php ${SCRIPT_DIR}/writeRfilev1.php "${DEPT_SRC}" "${GRILLE_SRC}" "${IMAGE_DIR}" "${KML_DIR}" "${ASCII_DIR}" "${R_FILE}" "${C_FILE}" &


#lancement d'une execution "R" sur le fichier R_FILE
# Rq. no save pour ne pas garder les variables en session, puis redirection des éventuelles erreurs dans le fichier de log
R --no-save -f ${R_FILE} 2>&1 >> $LOG &


dateFin=`date '+%y/%m/%d a %H:%M:%S'`
echo "${NUM_DEPT} terminé le ${dateFin} >> ${LOG_FILE}"

exit
```

Dans cet exemple de script .sh, il y a tout d'abord une définition de variables et chemins de fichiers. Le lancement d'un script php permet ensuite de générer dynamiquement du script "R" lui même exécuté avec la commande "R -- no-save -f ${R_FILE}".

Chapitre 2 : Calculs statistiques de base. Applications sur le logiciel “R”

L'objectif de ce chapitre vise dans un premier temps à d'aborder des calculs de base en statistiques (moyennes, variances, coefficient de corrélation...) tout en s'appropriant le logiciel R.

Dans un second temps, les chapitres 2.3, 2.4 et 2.5. seront consacrés à des analyses permettant de mesurer des relations entre variables.

On peut distinguer :

– a. les méthodes “explicatives” :

Ces méthodes visent à expliquer une ou des variables dites “dépendantes” (variables à expliquer) par un ensemble de variables dites “indépendantes” (variables explicatives).

- le cas des régressions (linéaires, polynomiales et coefficients associés).

– b. les méthodes “descriptives” :

Ces méthodes visent à structurer et résumer l'information par la transformation des variables que l'on cherche à comparer.

On abordera :

- les Analyses Factorielles des Correspondances ou AFC sur des données qualitatives (ce sont des tableaux de fréquences - ou contingence - qui sont utilisés en entrée).
- les Analyses en Composantes Principales ou ACP sur des données quantitatives (ce sont des tableaux de données quantitatives qui sont utilisés en entrée).

RQ. Il existe aussi les ACM (Analyses en Composantes Multiples) pour les AFC de plus de 2 variables.

– c. la comparaison de séries de données : les tests statistiques

- tests de dépendances : ex. Test t de student (Fisher), test du Khi2, test de U mann et whitney

2.1 Calculs de moyennes, écart-types, variances

– Principes:

- Moyenne :

Moyenne = Somme de valeurs quantitatives / Nb d'individus

- Variance :

La variance est une mesure servant à caractériser la dispersion d'un échantillon. Plus elle est faible, plus les valeurs sont regroupées autour de la moyenne. Elle correspond à la somme des différences au carré entre chaque valeur et la moyenne, divisée par l'effectif de l'échantillon.

$$\text{Variance} = \frac{\sum (x - \overline{x})^2}{n}$$

- Ecart-types :

L'écart type est directement dépendant de la variance. C'est la racine carré de la variance .

$$\text{Ecart-type} = \sqrt{\frac{\sum (x - \overline{x})^2}{n}}$$

– Exemples avec R:

Soit une série de 8 personnes avec les poids suivants : 60Kg, 56Kg, 61Kg, 68Kg, 51Kg, 53Kg, 69Kg, 54Kg.

Calculez la moyenne, l'ec-type et la variance avec R.

fonction **c()** : pour construire un vecteur de données
fonction **mean()** : pour calculer la moyenne à partir d'un vecteur de données
fonction **sd()** : pour calculer l'écart-type
fonction **var()** : pour calculer la variance

1. Créer un vecteur (vect1) : c(60, 56, 61, 68, 51, 53, 69, 54)

```
vect1 <-c(60, 56, 61, 68, 51, 53, 69, 54);
```

2. Créer la moyenne (moy1)

```
moy1 <- mean(vect1); = 59
```

3. Calcul Ecart-type

```
ec1 <- sd(vect1); = 6,761..
```

4. Calcul Variance

```
var1 <- var(vect1); = 45,714..
```

Remarque : La variance donnée par R est une variance non-biaisée avec dénominateur n-1 (estimation de la variance sur la base d'une population avec des effectifs inconnus. Cela permet “d'augmenter” la variance. Lorsque n devient grand, 1/(n-1) se rapproche de 1/n).

Le dénominateur n ne devrait être utilisé que pour le calcul d'une vraie variance c'est à dire lorsque l'on dispose de tous les individus d'une population (donc variance non estimée).

```
ex. Variance non biaisée
> var(vect1)
[1] 45.71429
>1/(length(vect1)-1)*sum((vect1 - mean(vect1))^2)
[1] 45.71429

ex. vraie variance
> 1/length(vect1)*sum((vect1 - mean(vect1))^2)
[1] 40
ex. vrai ec-type = 6,32
```

2.2 Calcul d'un coefficient de corrélation, covariances

– Principe :

Ce coefficient permet de mesurer **une relation linéaire potentielle** entre deux caractères quantitatifs continus. Le coefficient de corrélation linéaire (r) de deux caractères X et Y : C'est la covariance de X et Y divisée par le produit des écarts-types de X et Y. Plus la valeur de r se rapproche de +-1, plus la relation linéaire est forte et plus la valeur de r est voisine de 0, plus la relation est faible.

1. Calcul de la covariance :

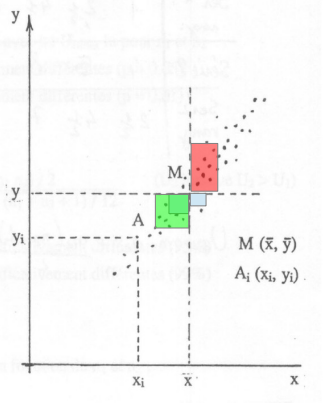
$$Cov(X,Y) = \frac{1}{N}\sum_{i=1}^{N}\left(X_i - \overline{X}\right).\left(Y_i - \overline{Y}\right)$$

En bref, plus la covariance est faible et plus les séries sont indépendantes. Inversement plus elle est élevée et plus les séries sont liées. Une covariance nulle correspondant à deux variables totalement indépendantes.

2. Calcul du coefficient de corrélation

La formule la plus simple utilisée est la suivante :

$$r(X,Y) = \frac{Cov(X,Y)}{\sigma_X.\sigma_Y}$$



En revanche la mesure de corrélation la plus utilisée est le coefficient de corrélation de Pearson. (c'est la formule qui est utilisée dans les fonctions lm() de R et CORREL d'Excel) :

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

où x est la moyenne de x et y la moyenne de y. La corrélation de Pearson fonctionne mieux avec des données qui suivent une distribution normale.

– Exemple :

Recherche d'une relation entre la taille des pieds des enfants et leur intelligence. Partant d'un échantillon de 10 enfants (notés A, B, ...J) on examinera s'il existe une corrélation linéaire entre la pointure de leurs chaussures (X) et leur quotient intellectuel (Y). Les données de l'analyse sont rassemblées dans le tableau 1, ci-dessous :

enfant (i)	Xi	Yi
A	31	50
B	31	55
C	32	52
D	33	56
E	33	63
F	34	65
G	35	69
H	36	90
I	37	110
J	38	150

Construction d'une matrice et calcul de la covariance puis du coefficient de corrélation avec R :

fonction **matrix()** : pour construire une matrice de données

fonction **cov()** : pour calculer la covariance à partir d'une matrice

fonction **cor()** : pour calculer le coefficient de corrélation

```
#Construction d'une matrice (par colonne ou par ligne)
mdatbyrow <- matrix(c(31,50,31,55,32,52,33,56,33,63,34,65,35,69,36,90,37,110,38,150), nrow = 10, ncol=2, byrow=TRUE);
mdatbycol <- matrix(c(31,31,32,33,33,34,35,36,37,38,50,55,52,56,63,65,69,90,110,150), nrow = 10, ncol=2, byrow=FALSE);

# calcul de la covariance
covbyRow  <-cov(mdatbyrow[,1], mdatbyrow[,2]);
covbyCol  <-cov(mdatbycol[,1], mdatbycol[,2]);

# calcul du coefficient de corrélation (Methode : Pearson)

corbyRow  <-cor(mdatbyrow[,1], mdatbyrow[,2]);
corbyCol  <-cor(mdatbycol[,1], mdatbycol[,2]);
```

Exemple de script permettant de calculer une covariance et la corrélation à partir de données saisies dans une matrice.

RQ. mdatbyrow et mdatbycol sont deux façons de construire la même matrice.

RQ. Il est possible d'appeler directement la première ou deuxième colonne en utilisant Tab[,numcol]. Pour appeller un numéro de ligne on utilise Tab[numLigne,]

Résultats :

```
> mdatbycol
      [,1] [,2]
 [1,]   31   50
 [2,]   31   55
 [3,]   32   52
 [4,]   33   56
 [5,]   33   63
 [6,]   34   65
 [7,]   35   69
 [8,]   36   90
 [9,]   37  110
[10,]   38  150
> covbyRow
[1] 71.22222
> covbyCol
[1] 71.22222
> corbyRow
[1] 0.9064746
> corbyCol
[1] 0.9064746
```

Les calculs donnés par la méthode de Pearson donnent pour covariance 71,22 et une corrélation de 0,90.

Le test de significativité (test d'hypothèse) par rapport à la table de Spearman, montre que pour 10 observations, les valeurs critiques du coefficient de Spearman sont de 0.56 au seuil de 5% et de 0.75 au seuil de 1%.

Table du Rho r de Spearman

n / a	0.05	0.01	n / a	0.05	0.01
4	1.00	-	24	0.34	0.49
5	0.90	1.00	26	0.33	0.47
6	0.83	0.94	28	0.32	0.45
7	0.71	0.89	30	0.31	0.43
8	0.64	0.83	35	0.28	0.40
9	0.60	0.78	40	0.26	0.37
10	0.56	0.75	45	0.25	0.35
12	0.51	0.71	50	0.24	0.33
14	0.46	0.64	55	0.22	0.32
16	0.42	0.60	60	0.21	0.30
18	0.40	0.56	70	0.20	0.28
20	0.38	0.53	80	0.19	0.26
22	0.36	0.51	100	0.17	0.23

n: le nombre d'observations. a : le risque d'erreur.

Dans notre cas (corrélation = 0.9 > 0.75), on peut affirmer avec moins de 1% de chance de se tromper que la relation observée n'est pas le fruit du hasard.

2.3 Les méthodes “explicatives”

Les méthodes explicatives vont chercher à mesurer des degrés de dépendance entre des varaible allant parfois jusqu'à prédire les valeurs d'une variable (calculs de régressions) en fonction des valeurs d'une autre variable.

2.3.a Régression linéaire

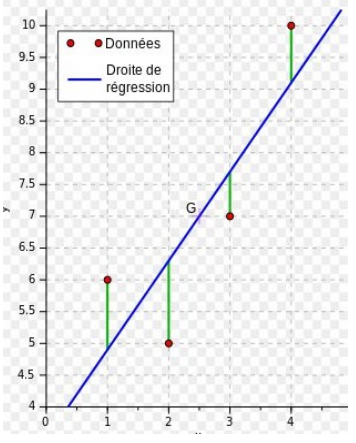
– Principe :

Calculer le coefficient de corrélation entre 2 variables numériques revient à chercher à résumer la liaison qui existe entre les variables à l'aide d'une droite. On parle alors d'un ajustement linéaire.

Soit une droite d'équation F(x) = a*x + b ;

Le **coeff. directeur “a”** = covariance(X,Y) / variance (X)

“b” est l'ordonnée à l'origine.



Le point moyen G (moy. X, moy. Y) ou barycentre fait partie de la droite, on connait donc au moins un point de cette droite. Ce qui va permettre de calculer b.

– Exemple :

Calcul de la droite de régression avec l'exemple des pointures de chaussures et Quotien Intellectuel (“QI”) avec R :

1. Exploitation de la matrice sous forme des variable X et Y :

Ici on veut exprimer le QI (valeur de la deuxième colonne) en fonction de la pointure des chaussures (valeurs de la secondes colonne)

```
y <- mdatbycol[,2]
x <- mdatbycol[,1]
```

2. Utilisation de la fonction **lm()** pour obtenir le coefficient directeur (a) et b

Pour connaitre les paramètres d'une fonction de premier degré de type Y=aX+b, il faut signifier à la fonction lm(), l'équation correspondante.

Pour une équation du premier degré il faut écrire :

lm(y~1 + x);

y~1 : correspond à b

x : correspond au coefficient directeur (a)

Exemple de script R avec génération de graphiques :

#construction des vecteurs

```
x <- c(31,31,32,33,33,34,35,36,37,38);
y <- c(50,55,52,56,63,65,69,90,110,150);
```

ou

```
y <- mdatbycol[,2]
x <- mdatbycol[,1]
```

#formule de regression lineaire (équation du premier degré)

```
reg <- lm(y~1+x);
```

#pour avoir le detail des coefficients obtenus : coef[1]=b et coef[2]=a pour y = ax+b on utilise le paramètre "$coef" :

```
reg$coef;
```

#pour avoir des informations sur la régression utiliser summary() :

```
summary(reg);
```

#eventuellement les résidus (écarts en ordonnée par rapport à la regression)

```
reg$residuals;
```

la construction d'un graphique va nécessiter :

- l'utilisation de la fonction **plot()** avec une définition des axes
- le calcul des coordonnées de la droite ou courbe de régression
- l'utilisation de la fonction **points()** pour positionner les points correspondant aux mesures initiales

#vecteur des abscisses : panel de valeurs de 0 à 200

```
vect2 <- 0:200;
```

#calcul des ordonnees (vect4) avec la regression en utilisant les valeurs de vect2

```
vect4 <- reg$coef[1] + reg$coef[2]*vect2;
```

#construction du graphique avec **plot()**

```
plot(vect2,vect4,type="l",col="red",xlim=c(0,50),ylim=c(0,160));
```

ajout des points avec la fonction **points()**

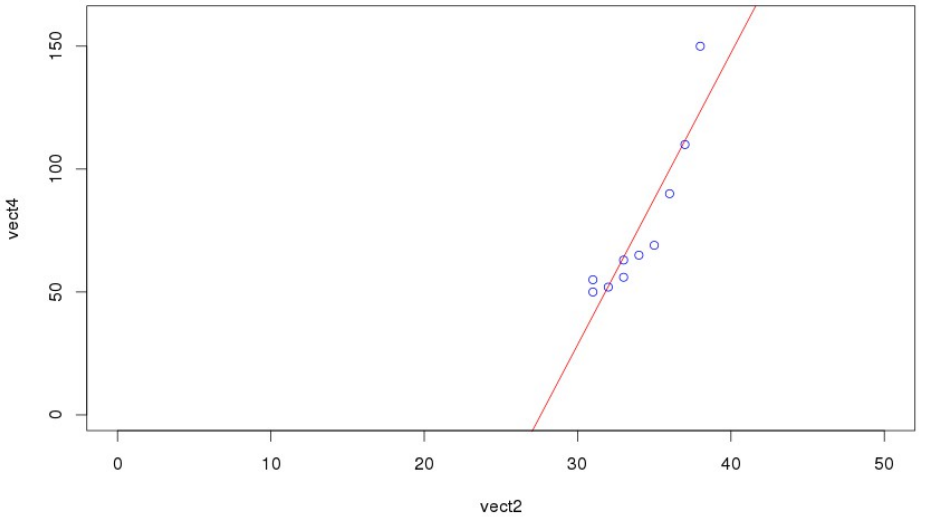
```
points(x,y,col="blue");
```

Rq : l'attribut' $**coef** ' d'une regression fournit le coefficient directeur (a) et b voire en plus les autres résidus pour les régressions polynomiales.

Rq : pour les fonctions de graphiques **plot()** et **points()** on utilise les paramètres :

- **type** = "l" ou "p" .. pour ligne ou point
- **col** = pour la coloration (en anglais)
- **xlim et ylim** = vecteur () : limite des axes sous forme de vecteurs

RQ. pour avoir davantage de détail sur la significativité de la fonction il est intéressant d'utiliser la fonction **summary().** Nous obtenons ainsi le coefficient de détermination (`R-squared`), qui est un bon indicateur de l'ajustement de la droite de régression.



Regression linéaire obtenue. Graphique mettant en relation les valeurs des pointures de chaussures et QI. La droite de régression linéaire est (F(x) = 11,87x – 327,59;)

Rq. En profiter pour modifier les paramètres de plot() : ex. modifier le type, ajouter un sous-titre, modifier la couleur, modifier les titre des axes.

Exemple:

```
plot(vect2,vect4,type="l",col="red",xlim=c(20,50),ylim=c(0,
160), sub="Droite de régression linéaire entre le QI et la
pointure des chaussures pour 10 individus",
xlab="Pointures", ylab="QI");
```

Résultat :

```
> summary(reg)

Call:
lm(formula = y ~ 1 + x)

Residuals:
    Min      1Q  Median      3Q     Max
-18.870  -9.338  -1.370   7.144  26.518

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -327.593     66.625  -4.917 0.001168 **
x             11.870      1.955   6.072 0.000299 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

Residual standard error: 14.37 on 8 degrees of freedom
Multiple R-squared: 0.8217,     Adjusted R-squared: 0.7994
F-statistic: 36.87 on 1 and 8 DF,  p-value: 0.0002986
```

En utilisant R, la droite de régression obtenue est donc F(x) = 11,87x - 327,59;
Le coefficient de détermination est d'environ 0,8217 (plutôt bon).

RQ. Le coefficient de détermination correspond au carré du coefficient de corrélation. On retrouve donc la valeur du coefficient de corrélation en recherchant la racine carré du coefficient de détermination :

```
sqrt(0.8217); = 0.9064747
```

2.3.b Régression polynomiale

– Principe :

La régression polynomiale est une analyse statistique qui cherche à décrire une variable aléatoire expliquée (Y) en fonction d'une variable aléatoire explicative (X) grâce à des fonctions complexes. De façon générale, plus on augmente le dégré de la fonction et plus on "ajuste" la courbe de régression aux valeurs fournies par une matrice de points.

On cherche par ce calcul à lier les variables par un polynôme de degré n.

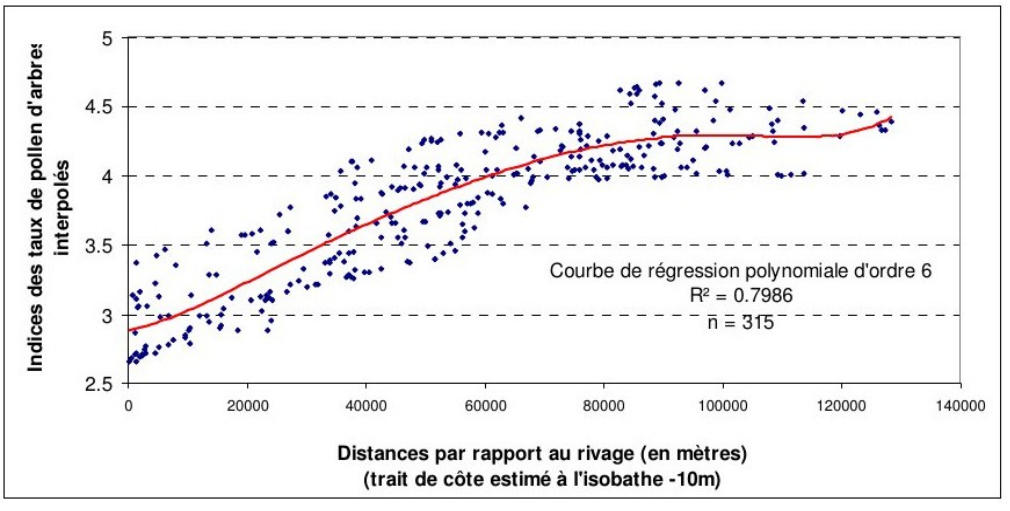
- Le calcul de la moyenne est une régression polynomiale de degré 0.

- La régression linéaire (**Y = a.X +b**) est une régression polynomiale de degré 1 (ou équation du premier degré).

- La fonction de second degré : $\mathbf{Y = a.x^2 + b.x + c}$ à une regression de degré 2.

- La régression linéaire de degré n s'écrira :

$$Pn(x) = a_n.x^n + a_{(n-1)}.X^{n-1} + \ldots + a1.x + a0$$



Exemple de regression polynomiale de degré 6

Rq. Il peut apparaitre intéressant d'augmenter le degré des équations des régressions polynomiales, néanmoins cela va augmenter grandement les ressources et le temps de calcul des processus. Cela peut constituer un inconvénient majeur dans le cadre de l'automatisation des calculs.

– Exemple :

Calcul de la courbe de régression de degré 2 avec l'exemple des pointures de chaussures et “Quotien Intellectuel” (QI) avec R :

Exemple de script R avec génération de graphiques :

#construction des vecteurs

```
x <- c(31,31,32,33,33,34,35,36,37,38)
y <- c(50,55,52,56,63,65,69,90,110,150)
```

#construction du vecteur x2 correspondant

```
x2 <- x*x
```

#construction de la formule de régression du second degré

```
reg2 <- lm(y~1+x+x2)
```

#vecteur des abscisses

```
vect2 <- 0:200
```

#vecteur des ordonnees en utilisant la fonction reg2

```
vect3 <- reg2$coef[1] + reg2$coef[2]*vect2+reg2$coef[3]*vect2*vect2
```

#construction des graphiques

```
plot(vect2,vect3,type="l",col="red",xlim=c(0,50),ylim=c(0,160));
points(x,y,col="blue");
```

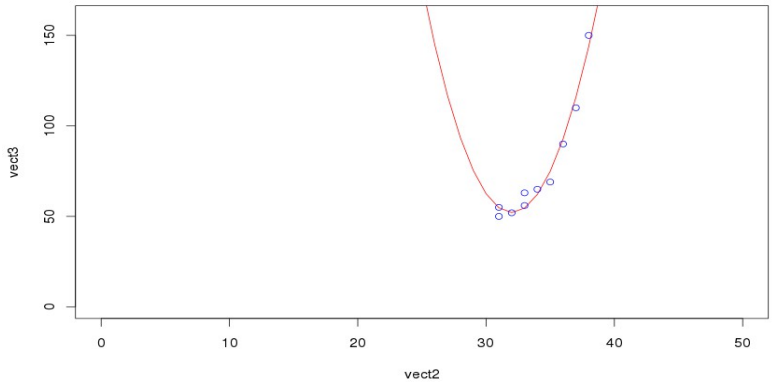
Résultat et génération de graphiques :

```
> summary(reg2);

Call:
lm(formula = y ~ 1 + x + x2)

Residuals:
    Min      1Q  Median      3Q     Max
-5.8867 -4.2386  0.0861  2.4878  8.4373

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2678.9928   442.6851   6.052 0.000515 ***
x           -164.0932    25.8756  -6.342 0.000388 ***
x2             2.5626     0.3767   6.803 0.000252 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
’ 1

Residual standard error: 5.567 on 7 degrees of freedom
Multiple R-squared: 0.9766,     Adjusted R-squared: 0.9699
F-statistic: 145.9 on 2 and 7 DF,  p-value: 1.967e-06
```

En utilisant la fonction summary sur reg2 nous obtenons la regression

$F(x) = 2{,}5626x^2-164{,}0932x+2678{,}99$

Le coefficient de détermination est meilleur que la regression linéaire autour de 0,97.



Regression du second degré mettant en relation les valeurs des pointures et QI. La droite de régression linéaire est $F(x) = 2{,}5626x^2-164{,}0932x+2678{,}99$

2.3.c Coefficient de détermination

– Principe :

Ce coefficient permet d'évaluer le degré d'association entre deux variables et de juger de la qualité de l'ajustement des points par rapport à une courbe ou une droite de régression.

R2 ou coefficient de détermination est le carré du coefficient de corrélation.

- En régression simple, un R2 proche de 1 est suffisant pour dire que l'ajustement est bon.

- En régression multiple, une valeur élevée du coefficient de détermination n'est pas suffisante, il est normalement nécessaire d'effectuer un test sur la significativité de R (test de Fisher).

RQ. En vue de faire des modélisations, il est souhaitable que la valeur du “R2” soit élevée.

– Exemple :

RQ. Le coefficient de détermination est fourni par la fonction **summary().** Nous obtenons ainsi le coefficient de détermination, qui est un bon indicateur de l'ajustement d'une courbe de régression.

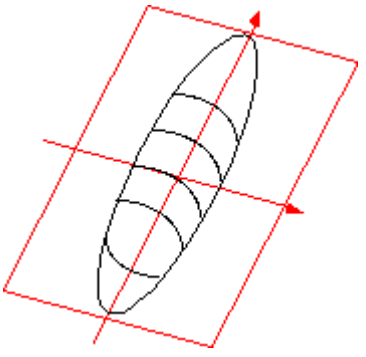
2.4 Les méthodes “descriptives” : cas des analyses factorielles

La présentation synthétique d’un grand ensemble de données résultant de l’étude de plusieurs caractères quantitatifs ou qualitatifs sur une population n’est pas facile.

Les analyses factorielles ont pour but de révéler les interrelations entre ces caractères et de proposer une structure de la population de **façon graphique** (descriptive).

Les méthodes factorielles cherchent **à réduire le nombre de variables en les résumant par un petit nombre de variables synthétiques** (variables “centrées et réduites”). Cela passe par la transformation de toutes les variables en unités compatibles : **ce sont les axes factoriels**.

Pour cela, on procède à des ajustements linéaires successifs du nuage initial (cf. régressions), afin de déterminer l'axe qui restitue le mieux la forme géométrique du nuage. C'est le premier axe d'inertie du nuage. Puis on établit le 2nd axe factoriel, et le plan factoriel F1 – F2 qui porte le maximum d'information (maximum de variance).



Exemple de plan factoriel F1-F2 associé à un nuage de points “multiparamétrés” dont le volume est de forme ovale.

Il existe les **AFC (Analyses Factorielles des Correspondances)** qui traitent de **données qualitatives** et les **ACP (Analyses en Composantes Principales)** qui traitent de **données quantitatives**.

- Exemple des AFC (traite de données qualitatives ou comptages) :

1. Utilisation d'un tableau de contingence :

Un tableau de contingence est un tableau de comptage permettant d'estimer la dépendance entre “deux caractères”. Ces caractères sont définis sous la forme de catégories permettant de caractériser une population. Les valeurs renseignées correspondent à des comptages effectués sur cette population.

Exemple 1 : Tableau de contingence où ce sont des sites (en quelque sorte des “objets” multiparamétrés) qui sont indiqués en ligne. Les caractères qualitatifs sont indiqués en colonnes. A noter ici que les deux premières colonnes n'entrent pas dans le calcul. Elles sont considérées comme des variables “illustratives”. Cas de sites caractérisés par des occurences d'espèces ou unité de végétation (Présence ou pas de Céréales, Vignes, Chanvre, forêts, bois clairs, etc...). (cf. Fichier `/Donnees/Donnees_Chapitre_4/Exercice1_AB/table_pour_AFC_vegetation_AB.ods);`

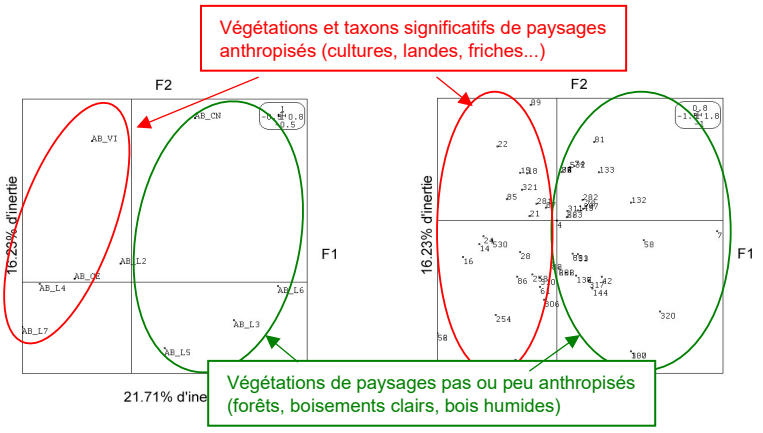
En ligne : liste d'échantillons
En colonne : caractères qualitatifs en Prés. /Abs. d'une espèce végétale à l'intérieur de chaque site (3 colonnes sur 9 sont représentées).

NOM_STRC	ID_ECH_F	P/A_CE	P/A_VI	P/A_CN
Villepail-La Corni	4	0	0	0
Blandouet - La B	7	0	0	0
Change - GlatiniU	8	1	0	0
Saint-Brevin - Le	14	0	0	0
Saint-Pere-en-Re	15	1	1	0
Haut-Clion - La F	16	1	0	0
Saint-Michel-Che	18	1	1	0
Saint-Viaud - Car	21	1	1	0
Saint-Cyr-en-Ret	22	1	1	0

L'AFC a ici deux objectifs :

1. Identifier des “groupes de végétations” à partir des végétaux qui sont souvent associés dans les sites.

2. Rassembler les sites qui ont des compositions végétales semblables.



La position des catégories de végétations dans le plan factoriel de gauche a permis d'associer l'axe F1 à un “degré d'anthropisation des paysages” : végétations anthropisées (cultures, friches, landes) à gauche et végétations de paysages naturels (forêts) à droite. Notons que l'axe F1 porte 21,71 % de la variance totale, l'axe F2 16%. Le plan factoriel de droite permet d'y associer les sites (identifiants des prélèvements).

Exemple 2 : Tableau de contingence dans lequel on a cherché à connaitre quel était la répartition des tâches quotidiennes au sein d'une famille, entre une femme et son époux. On a croisé deux caractères “qualitatifs” représentés par des catégories :

	Wife	Alternating	Husband	Jointly
Laundry	156	14	2	4
Main_meal	124	20	5	4
Dinner	77	11	7	13
Breakfeast	82	36	15	7
Tidying	53	11	1	57
Dishes	32	24	4	53
Shopping	33	23	9	55
Official	12	46	23	15
Driving	10	51	75	3
Finances	13	13	21	66
Insurance	8	1	53	77
Repairs	0	3	160	2
Holidays	0	1	6	153

En ligne : les différentes tâches domestiques possibles

En colonne : la valeur indiquée représente la fréquence (comptages) d'executions de plusieurs situations possibles ou plusieurs catégories de personnes : ex. épouse seule (wife), de façon alternée (alternating), époux seul (husband) et les deux en même temps.

Dans cet exemple on attend de l'AFC :

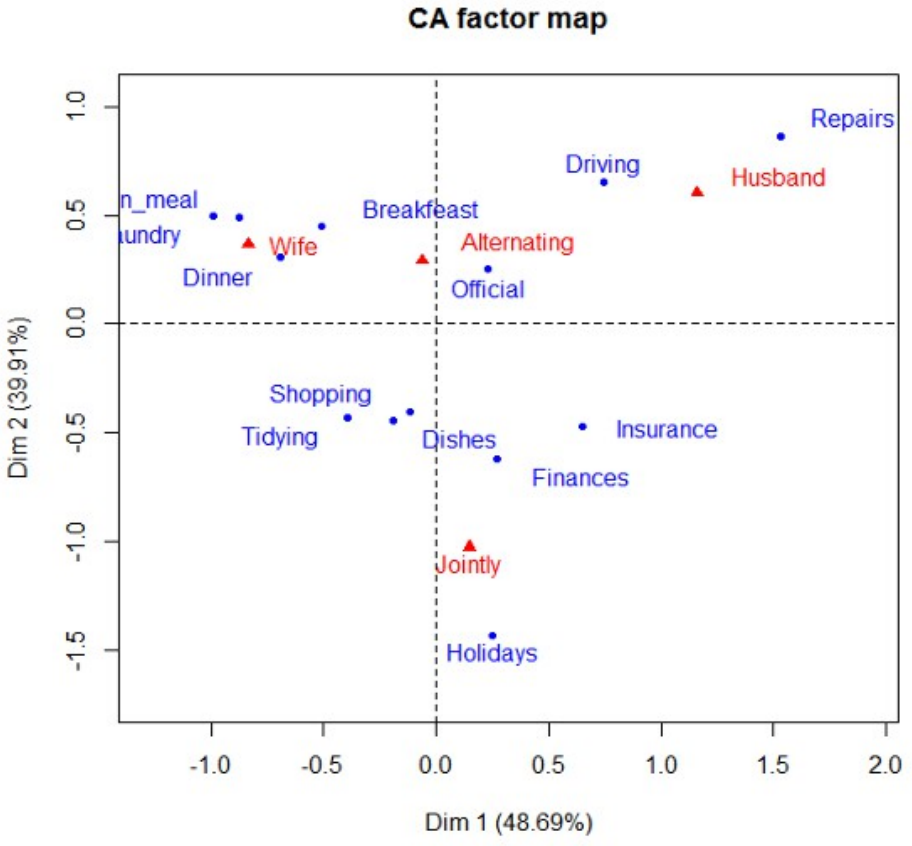
1. de rassembler les tâches domestiques qui sont souvent réalisées de la même façon par les mêmes catégorie de personnes,

2. de pouvoir visualiser l'association des tâches domestiques aux catégories de personnes.

2. Représentation des catégories des deux variables grâce à une Analyse Factorielle des Correspondances : obtention du plan factoriel (F1-F2) sous R :

```
#utilisation de la librairie FactoMineR
library(FactoMineR)

dataAFC<-
read.csv(file="/donnees/Donnees_Chapitre_2/afc_tache_menage
res.txt",sep="\t", header=TRUE, row.names=1,
check.names=FALSE);

#réalisation de l'AFC
res.ca = CA(dataAFC)
```

```
#La commande dessine automatiquement le plan factoriel des
variables colonnes et lignes.
```



Plan factoriel des correspondances F1-F2. Les catégories des deux variables sont superposées dans le plan.

L'axe F1 décrit les tâches à fort “contraste” entre l'homme et la femme : les préparations des repas, le linge sont le domaine des femmes alors que le bricolage ou la conduite sont le domaine des hommes.

L'axe F1 est le plus explicatif dans le tri des taches (48% d'explication)

L'axe F2 permet de distinguer les tâches conjointes.

```
# Affiche les valeurs propres/contributions des axes
factoriels
res.ca$eig
```

	valeurs propres ou variances	% variance	% cumulés
dim 1	0.542889285	48.69222124	48.69222124
dim 2	0.445002761	39.91269215	88.60491338
dim 3	0.127048433	11.39508662	99.99
dim 4	5.12E-33	4.59E-31	100

Valeurs propres ou variances calculées pour chaque axe (ou composante)

Pour aller plus loin :

La construction des nouveaux axes par “centrage -réduction” donne lieu à des plans factoriels qui permettent d'obtenir une visualisation des données dans des plans où les valeurs des différentes variables ont les mêmes unités. C'est le propre des méthodes descriptives.

Les coordonnées des données (les tâches domestiques en fonction des catégories de personnes) sont recalculées et projetées dans ces plans. Connaissant les nouvelles coordonnées, il est possible de calculer une variance – ou valeur propre – pour chaque axe.

Les axes qui présentent le plus de “variabilité” (ou variance) sont considérés comme les plus importants car ce sont ceux qui permettent “d'élargir ou de dilater” le plus le nuage de points projetés.

Il peut être pratique de calculer le pourcentage de variance pour chaque axe, car on peut alors attribuer une part d'explication pour chaque composante. Il est aussi possible de cumuler les pourcentages pour connaitre la part totale de l'explication pour les 2, 3, 4... premiers axes.

Dans notre exemple, “F1-F2” représente 88,6% (48,69% + 39,91%) de la variance ce qui est très bien. Il reste une part de l'explication de la variabilité de l'ordre de 12%, relativement négligeable, correspondant à l'axe 3. Selon les situations, il peut être intéressant de regarder le plan “F1-F3”.

```
# contribution des modalités colonnes aux axes
factoriels
res.ca$col$contrib
```

	Dim 1	Dim 2	Dim 3
Wife	44.462018	10.312237	10.8220753
Alternating	0.103739	2.782794	82.5492464
Husband	54.233879	17.786612	6.1331792
Jointly	1.200364	69.118357	0.4954991

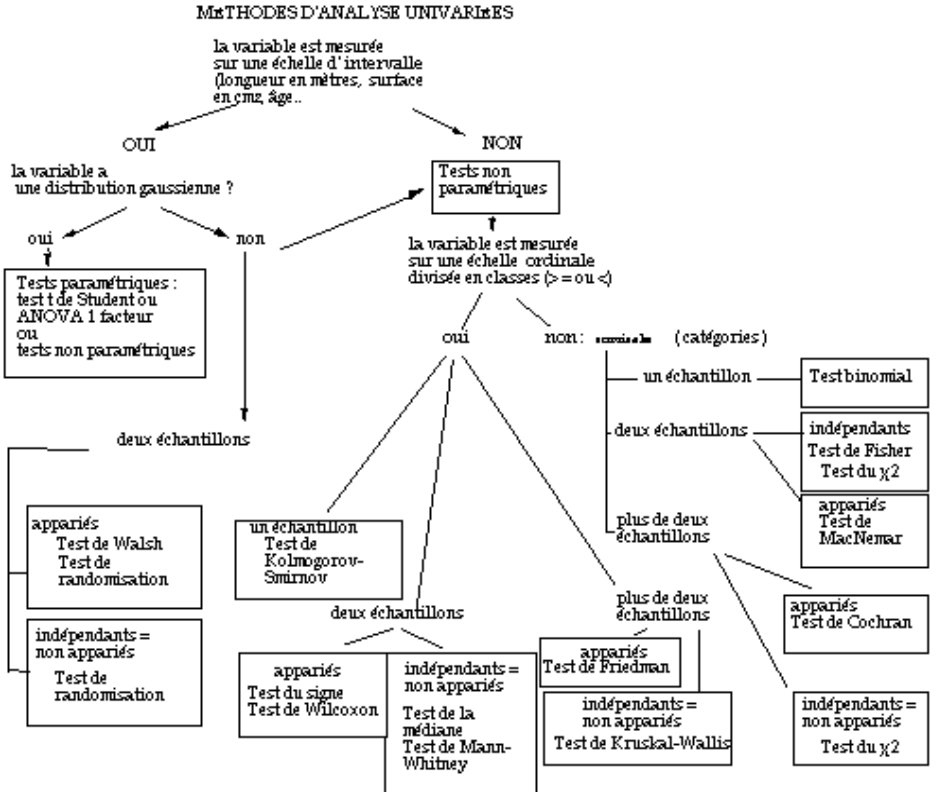
Les calculs peuvent aussi être faits catégorie par catégorie ce qui permet d'estimer la contribution des catégories par rapport aux axes. Résultats intéressants pour l'interprétation des résultats.

2.5 La comparaison de séries de données : les tests statistiques

Il s'agit de savoir si les séries de valeurs sont significativement différentes d'un point de vue statistique. Généralement, la démarche consiste à rejeter - ou pas – une hypothèse nulle, en fonction d'un jeu de données.

RQ. Dans les tests suivants nous nous intéresserons uniquement à des comparaisons de deux variables à la fois.

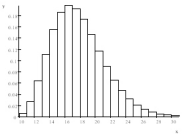
Le choix des outils statistiques à mettre en place dépend en règle générale du type de donnée ou mesure (donnée qualitative ou quantitative), de la forme de la distribution (loi normale ou pas) et du nombre d'échantillons dont on dispose. (voir schéma).



Les cas de figures sont inommbrables. Afin de ne pas trop alourdir la formation nous nous sommes intéressés aux principaux cas de figures et tests.

- Principaux outils à disposition pour comparer des deux séries de données :

1. Comparaison de séries de valeurs quantitatives :



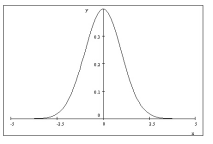
Probability of 10 successes in x trials, given probability 0.6 of success in each trial.

- Si on ne connait pas le type de distribution des donnée ou si les séries de valeurs ne sont pas de type normale (Variance > Moyenne) :

C'est à dire si l'on a une forte variabilité des valeurs.

→ Utilisation de tests non paramétriques : ex. Test U de Mann et Whitney ou Wilcoxon

- Si les séries sont de type normale (Variance < Moyenne) :



→ Utilisation du test t de student : test paramétrique car dépend de la moyenne et de la variance, généralisé par Fisher.

- Si l'on souhaite comparer une série expérimentale (effectif) et une série théorique :

→ Utilisation du test du Khi2, (test de conformité)

2. Comparaison de séries de valeurs qualitatives :

→ Utilisation du test du Khi2 (test d'indépendance), non paramétrique.

Dans ce cas, on dispose d'une population où **chaque individu est repéré par deux caractères qualitatifs A et B.** (cf. Exercice, caractère A : le sexe et caractère B : le salaire) et on cherche à savoir si les caractères A et B sont indépendants ou non.

Exemple du test t de student :

Le **test-t de Student** est un **test statistique** permettant de **comparer les moyennes** de deux groupes d'échantillons. Il s'agit donc de savoir si les moyennes des deux groupes sont significativement différentes au point de vue **statistique**.

Il est utilisé pour comparer deux moyennes ou pour comparer une moyenne observée m à une valeur théorique mu. En fonction de cela, il existe différentes variantes du test de student :

-> Le test de student pour échantillons unique (ou one-sample t-test en anglais), utilisé pour comparer une moyenne observée à une valeur théorique.

-> Le test de student non-apparié, utilisé pour comparer les moyennes de deux groupes d'échantillons indépendants (ex. Les moyennes proviennent de deux populations différentes).

-> Le test de student apparié, utilisé pour comparer les moyennes de deux séries appariés (ex. les deux séries de tests sont effectuées sur une même population, par exemple avant et après une situation).

Sous R, la fonction à utiliser pour faire le test-t de student est t.test(). Elle permet de faire les différents types du test de student mentionnés ci-dessus. Un format simplifié de la fonction est montré ci-dessous :

```
# Comparaison d'une moyenne observée
# à une moyenne théorique mu
t.test(x, mu=0)

# Test de student non-apparié
# Comparaison des moyennes de deux groupes indépendants(x
et y)
t.test(x, y)

# Test de student apparié : permet de comparer la moyenne
de deux séries de valeurs ayant un lien
t.test(x, y, paired=TRUE)
```

Cas du Test t de student pour échantillons indépendants :

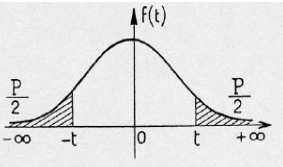
Il s'agit de comparer deux moyennes m1 et m2 afin de savoir si leur différence d=|m1-m2| est due au hasard.

L'hypothèse avancée est que les 2 échantillons appartiennent à la même distribution de type normale.

Le test consiste à calculer une valeur **|t|** qui va dépendre de la différence |m1-m2| et des variances des deux séries.

On compare ensuite la valeur **|t|** à une valeur T_{table} définie à partir d'une table (T95 ou T90 correspondant à des coefficient de sécurité = risque) pour un certain degré de liberté.

Si la valeur absolue de t (|t|) **est supérieure à la valeur critique** T_{table}**, alors la différence est significative**. Dans le cas contraire, elle, ne l'est pas. La fonction sur R donne directement une "**p-value**" ou **degré de significativité. Cette valeur** est à comparer au risque choisi par défaut pour définir la **table de Student**. Par défaut les valeurs de T_{table} retenues dans R sont construites avec un coefficent de sécurité de 95% (soit 5% de risque). **La p-value** est donc à comparer au risque de 5% soit la valeur 0,05.



ν \ P	0,90	0,80	0,70	0,60	0,50	0,40	0,30	0,20	0,10	0,05	0,02	0,01	0,001
1	0,158	0,325	0,510	0,727	1,000	1,376	1,963	3,078	6,314	12,706	31,821	63,657	636,619
2	0,142	0,289	0,445	0,617	0,816	1,061	1,386	1,886	2,920	4,303	6,965	9,925	31,598
3	0,137	0,277	0,424	0,584	0,765	0,978	1,250	1,638	2,353	3,182	4,541	5,841	12,929
4	0,134	0,271	0,414	0,569	0,741	0,941	1,190	1,533	2,132	2,776	3,747	4,604	8,610
5	0,132	0,267	0,408	0,559	0,727	0,920	1,156	1,476	2,015	2,571	3,365	4,032	6,869
6	0,131	0,265	0,404	0,553	0,718	0,906	1,134	1,440	1,943	2,447	3,143	3,707	5,959
7	0,130	0,263	0,402	0,549	0,711	0,896	1,119	1,415	1,895	2,365	2,998	3,499	5,408
8	0,130	0,262	0,399	0,546	0,706	0,889	1,108	1,397	1,860	2,306	2,896	3,355	5,041
9	0,129	0,261	0,398	0,543	0,703	0,883	1,100	1,383	1,833	2,262	2,821	3,250	4,781
10	0,129	0,260	0,397	0,542	0,700	0,879	1,093	1,372	1,812	2,228	2,764	3,169	4,587
11	0,129	0,260	0,396	0,540	0,697	0,876	1,088	1,363	1,796	2,201	2,718	3,106	4,437
12	0,128	0,259	0,395	0,539	0,695	0,873	1,083	1,356	1,782	2,179	2,681	3,055	4,318
13	0,128	0,259	0,394	0,538	0,694	0,870	1,079	1,350	1,771	2,160	2,650	3,012	4,221
14	0,128	0,258	0,393	0,537	0,692	0,868	1,076	1,345	1,761	2,145	2,624	2,977	4,140
15	0,128	0,258	0,393	0,536	0,691	0,866	1,074	1,341	1,753	2,131	2,602	2,947	4,073
16	0,128	0,258	0,392	0,535	0,690	0,865	1,071	1,337	1,746	2,120	2,583	2,921	4,015
17	0,128	0,257	0,392	0,534	0,689	0,863	1,069	1,333	1,740	2,110	2,567	2,898	3,965
18	0,127	0,257	0,392	0,534	0,688	0,862	1,067	1,330	1,734	2,101	2,552	2,878	3,922
19	0,127	0,257	0,391	0,533	0,688	0,861	1,066	1,328	1,729	2,093	2,539	2,861	3,883
20	0,127	0,257	0,391	0,533	0,687	0,860	1,064	1,325	1,725	2,086	2,528	2,845	3,850
21	0,127	0,257	0,391	0,532	0,686	0,859	1,063	1,323	1,721	2,080	2,518	2,831	3,819
22	0,127	0,256	0,390	0,532	0,686	0,858	1,061	1,321	1,717	2,074	2,508	2,819	3,792
23	0,127	0,256	0,390	0,532	0,685	0,858	1,060	1,319	1,714	2,069	2,500	2,807	3,767
24	0,127	0,256	0,390	0,531	0,685	0,857	1,059	1,318	1,711	2,064	2,492	2,797	3,745
25	0,127	0,256	0,390	0,531	0,684	0,856	1,058	1,316	1,708	2,060	2,485	2,787	3,725
26	0,127	0,256	0,390	0,531	0,684	0,856	1,058	1,315	1,706	2,056	2,479	2,779	3,707
27	0,127	0,256	0,389	0,531	0,684	0,855	1,057	1,314	1,703	2,052	2,473	2,771	3,690
28	0,127	0,256	0,389	0,530	0,683	0,855	1,056	1,313	1,701	2,048	2,467	2,763	3,674
29	0,127	0,256	0,389	0,530	0,683	0,854	1,055	1,311	1,699	2,045	2,462	2,756	3,659
30	0,127	0,256	0,389	0,530	0,683	0,854	1,055	1,310	1,697	2,042	2,457	2,750	3,646
40	0,126	0,255	0,388	0,529	0,681	0,851	1,050	1,303	1,684	2,021	2,423	2,704	3,551
80	0,126	0,254	0,387	0,527	0,679	0,848	1,046	1,296	1,671	2,000	2,390	2,660	3,460
120	0,126	0,254	0,386	0,526	0,677	0,845	1,041	1,289	1,658	1,980	2,358	2,617	3,373
∞	0,126	0,253	0,385	0,524	0,674	0,842	1,036	1,282	1,645	1,960	2,326	2,576	3,291

Table t de student : les lignes correspondent aux nombres de degrés de liberté, les coefficients de sécurité (risques) sont en colonnes.

Exercice : On souhaite comparer les poids moyens chez les hommes et les femmes, les données utilisées sont celles du fichier data_t_student.txt. Les deux séries sont indépendantes.

Question : le poids moyen des femmes est il significativement différent de celui des hommes?

id	Group	Poids
1	Femme	42.10
2	Femme	53.80
3	Femme	30.00
4	Femme	45.80
5	Femme	57.70
6	Femme	59.20
7	Femme	82.40
8	Femme	66.20
9	Femme	66.90
10	Femme	51.20
11	Homme	80.70
12	Homme	85.10
13	Homme	88.60
14	Homme	81.70
15	Homme	69.80
16	Homme	79.50
17	Homme	107.20
18	Homme	69.30
19	Homme	80.90
20	Homme	63.00

```
# import des données

data_t_student<-
read.csv(file="/Donnees/Donnees_Chapitre_2/data_t_student
.txt",sep=";", header=TRUE);

# calcul des deux moyennes (pour information)

moy_femme <-mean(data_t_student[1:10, 3])
moy_homme <-mean(data_t_student[11:20, 3])

# création des 2 vecteurs de données :

poids_femme <-(data_t_student[1:10, 3])
poids_homme <-(data_t_student[11:20, 3])

# Lancement du test t de student :
res<-t.test(poids_femme,poids_homme)
```

```
# Resultat

        Welch Two Sample t-test

data:  poids_femme and poids_homme
t = -4.1493, df = 17.448, p-value = 0.0006388
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -37.76246 -12.33754
sample estimates:
mean of x mean of y
    55.53     80.58
```

Dans le résultat ci-dessus :

Welch two Sample t-test : signifie que le test de Welch a été lancé avant le test t student. Ce test tient compte du fait que les variances ne sont pas supposées être égales dans l'option « par défaut » du test t utilisé par R, ce qui représente l'approche la plus prudente possible. Il peut être désactivé en précisant que les variances sont supposées être égales, mais il faut alors le vérifier préalablement..

(res<-t.test(`poids_femme,poids_homme, var.equal=TRUE`)).

t est la statistique de student (t = -4.1493), c'est cette valeur que l'on comparerait avec les valeurs de la table t de student, si l'on procédait de "façon manuelle". RQ. Dans la table t de student, pour un risque de 5% et 18 degrés de liberté, la valeur est de 2,101. La valeur absolue du test (t = 4,1493) est donc supérieure à la valeur de la table t.

df est le nombre de degrés de liberté (df= 17.448), (#nb éch. homme + nb éch. Femme - 2 catégories à comparer) : il permet de prendre en compte le nombre d'échantillons dans le calcul du test. A noter que le test de Welch a ici transformé légèrement le degré de liberté attendu (#18). C'est normal, on a fait une hypothèse supplémentaire (l'égalité des variances) qui permet de gagner de la puissance.

p-value est le degré de significativité du test (p-value = 0.0006388), à comparer avec le coefficent de sécurité de la table de t de student (par défaut c'est une probabilité de 95% qui est retenu soit un risque de 0,05). **Ici la p-value est largement inférieure à 0,05, la différence entre les deux séries est significative avec un coefficient de sécurité de 95%**.

L'hypothèse "alternative" (H1) est que les moyennes sont différentes (sous entendu : l'hypothèse nulle Ho est "les moyennes sont égales")

L'intervalle de confiance de la différence des moyennes à 95% est également montrée (intervalle de confiance= [-37,76246, -12,33754]); Pour rappel la différence des moyennes est de l'ordre de 55,53 – 80,58 = -25,05.

Enfin, on a la **valeur moyenne** des deux groupes (poids moyen des femmes = 55.53, poids moyen des hommes =80.58).

Exemple du test du Khi2 (ou Chi2 ou X^2) :

Ce test permet de comparer des répartitions d'effectifs.

On distingue :

- **le Khi2 de conformité** qui permet de comparer un effectif à une valeur théorique attendue,

- **les Khi2 d'homogénéité et d'indépendance** qui permettent de voir si un effectif peut être dû au seul hasard.

Le Khi2 de conformité permet de savoir si il y a une correspondance entre la théorie et une répartition observée. Le test du Khi-deux permet donc de voir si un échantillon est conforme à la théorie ou s'il en diffère significativement.

- L'hypothèse nulle (H0) est : la théorie diffère de l'observation. Le résultat du Khi² de conformité permet de rejeter ou non H0.
- **La p-value donne la probabilité de non-validation de H0 - la probabilité de voir une conformité entre la théorie et l'observation. Plus p-value est petite, plus la théorie et l'observation diffèrent.**

Le Khi2 d'indépendance permet de savoir si il y a indépendance entre 2 critères susceptibles de créer une différence de répartition.

Ce test est utilisé dans le cas où l'on dispose d'une population où **chaque individu est repéré par deux caractères qualitatifs A et B.** (cf. Exercice, caractère A : le sexe et caractère B : le salaire avec 4 modalités) et on cherche à savoir si les caractères A et B sont indépendants ou non.

Par exemple : est-ce que le fait de connaître la couleur des yeux de quelqu'un me permet de supposer sur son sexe ? Réponse : non.

Par exemple : est-ce que le fait de connaître la taille de quelqu'un permet de supposer sur son sexe ? Réponse : oui, plus un individu est petit, plus il y a des chances que ce soit une femme.

- L'hypothèse nulle (H0) est : le fait de connaître l'appartenance d'un individu à une population (selon un critère) ne donne aucun indice sur la caractéristique qui le défini selon l'autre critère.
- La p-value donne la probabilité de validation de H0 - la probabilité de ne voir aucun lien entre les critères. **Plus p-value est petite, plus il y a un lien entre les critères (et donc pas d'indépendance).**
- X-square, cette valeur classique renvoyée par un test de Khi2 permet de retrouver manuellement la p-value en s'aidant d'un tableau disponible dans un livre de statistiques.

Exercices :

Cas d'un test du Khi2 de conformité :

On effectue un croisement en génétique. En théorie, on devrait observer dans la descendance : 75% d'individus à yeux rouges et 25% à yeux blancs. On observe en réalité sur une génération de 39 individus : 32 individus à yeux rouges et 7 à yeux blancs.

Phénotype	Nombre d'individus	Proportions
Yeux rouges	32	82,00%
Yeux blancs	7	18,00%

Effectifs observés, issus d'un croisement génétique de lapins. Effectif total = 39.

Phénotype	Nombre d'individus	Proportions
Yeux rouges	29	75,00%
Yeux blancs	10	25,00%

Effectifs théoriques sur la base de 39 individus et de fréquences théoriques.

Objectif du test de Khi2 de conformité : vérifier si les les résultats que l'on observe diffèrent significativement de ce que la théorie laissait envisager.

On part sur l'hypothèse nulle (H0) : les résultats observés sont conformes à la théorie. Si l'hypothèse nulle est rejetée, alors on a un cas particulier de génétique qu'il faudra élucider.

Réalisation d'un test Khi2 de conformité avec le logiciel R project :

```
# Construction d'un vecteur de “descendance observée”
descendance <- c(7,32)
# probabilités théoriques : d'un point de vue théorique,
on devrait avoir 3/4 et 1/4 soit 75 et 25%
proba <- c(0.25,0.75)
# Réalisation du test de khi-deux
chisq.test(descendance,p=proba)

# Resultats
#     Chi-squared test for given probabilities
# data:  descendance
# X-squared = 1.0342, df = 1, p-value = 0.3092
```

La valeur du Khi2 est de 1,0342 pour 1 degré de liberté. De façon “manuelle”, il faut ensuite regarder les valeurs d'une table $Khi2_{table}$ exprimant des valeurs en fonction du nombre de degrés de liberté et pour plusieurs coefficients de sécurité. Exemple à 95% (pvalue=0.05), 99%(p=0.01), 99,5%(p=0.005).

df \ pvalue	0.995	0.99	0.975	0.95	0.9	0.1	0.05	0.025	0.01	0.005
1	0.000	0.000	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.86
5	0.412	0.554	0.831	1.145	1.61	9.236	11.07	12.833	15.086	16.75
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.69	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.18	2.733	3.49	13.362	15.507	17.535	20.09	21.955
9	1.735	2.088	2.7	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.94	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.92	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.3
13	3.565	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688	29.819
14	4.075	4.66	5.629	6.571	7.79	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.0	34.267
17	5.697	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.39	10.865	25.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.26	9.591	10.851	12.443	28.412	31.41	34.17	37.566	39.997

Tables montrant les valeurs de $Khi2_{table}$ pour différents coefficients de sécurité.

Pour l'interprétation on s'inspirera des comparaisons suivantes :

- Si X^2calculé $<X^{95}$, il existe plus de 5 chances sur 100 pour qu'il existe une valeur X^2 supérieure au X^2calculé. Ce risque ne peut être pris, **ces différences ne sont pas significatives.**
- Si $X^{95}<X^2$calculé $<X^{99}$, il existe entre 5 et 1 chance sur 100 pour qu'il existe une valeur X^2 supérieure au X^2calculé. Ce risque peut être pris, **ces différences sont significatives.**
- Si X^2calculé $>X^{99}$, il existe moins d'une chance sur 100 pour qu'il existe une valeur X^2 supérieure au X^2calculé. Ce risque peut être pris, **la différence est très significative**.

Dans notre exemple, X^2calculé = 1,0342 pour 1 degré de liberté. Le X^{95} pour ddl=1 est de 3,841. X^2calculé $<X^{95}$, l**es différences observées ne sont pas significatives.**

On récupère aussi directement la probabilité d'avoir une telle situation p-value = 0.3092 ==> Plus de 30% de situations où l'hypothèse serait rejetée à tort. Evenement similaire à H0: les résultats observés sont conformes à la théorie, il n'y a pas de différences significatives.

Cas d'un test du Khi2 d'indépendance :

Intéressons-nous aux salaires des hommes et des femmes. Imaginons que l'on a demandé à 290 hommes et 285 femmes leurs salaires.

Salaires	1000 à 2000	2000 à 3000	3000 à 4000	4000 à 5000	Total
Hommes	50	70	110	60	290
Femmes	80	75	100	30	285
Total	110	145	210	110	575

Répartition des salaires suivant deux catégories : hommes / femmes

Objectif du Khi-deux : vérifier si les hommes et les femmes ont effectivement le même salaire (hypothèse nulle H0) ou si, au contraire, leurs salaires diffèrent.

Hypothèse nulle (H0) : le fait de connaître le sexe ne permet pas d'aider à deviner la tranche salariale d'un d'individu et inversement.

Si l'hypothèse nulle est rejetée, alors on a une relation sexe-salaire qui indiquera ici que les femmes sont moins bien rémunérées.

Réalisation d'un test Khi2 d'indépendance avec le logiciel R project :

```
# Créations des vecteurs correspondant aux 2 catégories :
hommes = c(50,70,110,60)
femmes = c(80,75,100,30)
# Création d'une matrice comparative :
tableau = matrix(c(hommes, femmes),2,4,byrow=T)
# (2 : nombre de lignes et 4 nombres de colonnes
(tranches salariales))
# Réalisation du test khi-deux - les résultats sont
sauvegardés dans "khi_test"
khi_test = chisq.test(tableau)
khi_test # affiche le résultat du test


Pearson's Chi-squared test
data:  tableau
X-squared = 17.53, df = 3, p-value = 0.0005499
```

La p-value obtenue est très petite. Il y a donc un lien statistique entre le sexe et la tranche salariale.

Si nous reprenons le tableau des valeurs du X^2, X^{99} avec un ddl =3 est de l'ordre de 11,345.

Ainsi, X^2calculé > X^{99}, il existe moins d'une chance sur 100 pour qu'il existe une valeur X^2 supérieure au X^2calculé. Ce risque peut être pris, **la différence est très significative.**

Exemple du test de U Mann-Whitney :

Le test U de Mann-Whitney (aussi appelé test de la somme des rangs de Wilcoxon ou plus simplement test de Wilcoxon) sert à tester l'hypothèse selon laquelle la distribution des données est la même pour deux groupes. **C'est un test non-paramétrique de comparaison de moyennes de deux échantillons indépendants ou appariés.**

La p-value associée à ce test va ainsi répondre à la question suivante :

Si les données pour deux groupes étaient issues d'une même population, quelle serait la probabilité que l'on observe par hasard une différence de rangs (ou de distribution) entre les deux groupes aussi forte que celle observée sur les données? **Cette probabilité est donnée par la valeur p-value, à comparer au seuil de 0,05 ou (5%).**

Si p-value est supérieure à 5% alors la probabilité que l'on observe par hasard une différence de rangs (ou de distribution) aussi forte que celle observée sur les données est grande. On ne peut donc rejeter l'hypothèse que les deux séries de données (groupes) appartiennent à la même population.

Rappel : Le test U de Mann-Whitney est souvent utilisé comme solution alternative à l'utilisation d'un **test de Student (t-test)** dans le cas où les données **ne sont pas distribuées selon une loi normale** et/ou dans le cas où les données **sont peu nombreuses**. Il s'agit en effet d'un **test non-paramétrique**, c'est à dire un test qui ne repose pas sur une hypothèse de distribution des données.

Réalisation d'un test de U Mann-Whitney avec le logiciel R :

Soient x et y deux échantillons indépendants à comparer. La commande à utiliser est la suivante :

```
x <- c(0.80, 0.83, 1.89, 1.04, 1.45, 1.38, 1.91, 1.64,
0.73, 1.46)
y <- c(1.15, 0.88, 0.90, 0.74, 1.21)
wilcox.test(x, y)
# Résultat de la commande :
Wilcoxon rank sum test

data: x and y
W = 35, p-value = 0.2544
alternative hypothesis: true location shift is not equal to 0
```

Dans cet exemple la p-value est p=0,2544. La p-value est plus grande que 0.05: on ne parvient pas à rejeter l'hypothèse selon laquelle les données des deux groupes proviennent d'une même distribution au seuil $\alpha = 0,05$

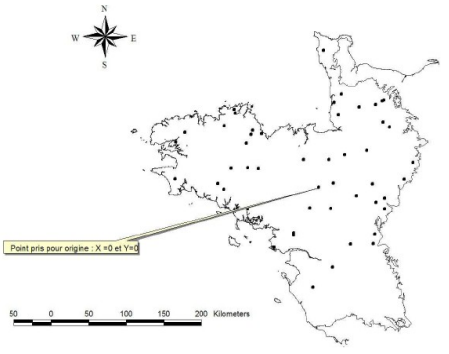
Les deux échantillons x et y ne sont pas significativement différents au seuil de 5% et la différence observée est peut-être liée au hasard.

Chapitre 3 : Introduction aux principales méthodes d'interpolation : méthode des Splines, Ponderation par l'Inverse de la distance (IDW), Krigeage. Exemples avec “R”.

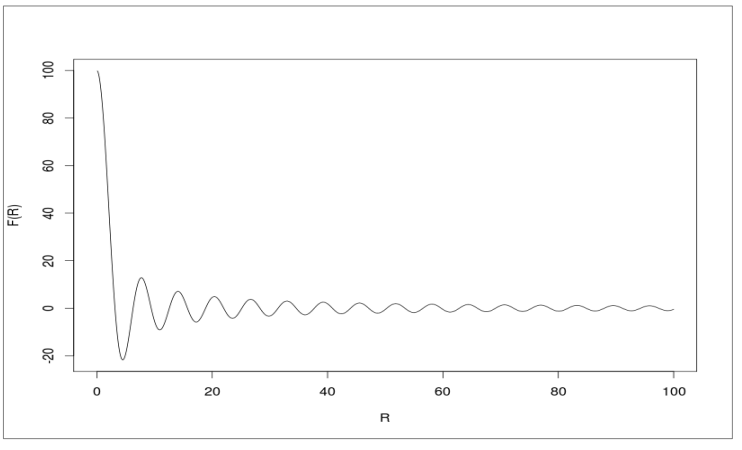
Comparaison des méthodes d'interpolation :

Afin de comparer les différentes méthodes nous sommes partis d'un nuage de points réparti de façon aléatoire à la surface du Massif armoricain.

Un point d'origine (x=0, y=0) a été placé au centre de la carte et une valeur (z) est calculée pour chacun des points avec la formule **$F(R) = 100 \text{ Sin } R/R \text{ avec } R=(x^2 + Y^2)^{1/2}x \text{ Pi}$**
RQ. Le pt d'origine (x=0 et y=0) étant placé au centre du raster :



RQ. Connaissant les coordonnées d'un pt, on peut calculer une valeur avec la formule du graphique de $F(R) = 100 \text{ Sin } R / R$ avec $R=((x^2 + Y^2)^{1/2})x \text{ Pi}$



ex. tableau de coordonnées avec le calcul :

D2 =RACINE(PUISSANCE((B2-296511,2304);2)+PUISSANCE((C2-2341309,038);2))

	A	B	C	D	E	F
1	UNIQUE_ID	RND_X_CRD	RND_Y_CRD	Rayon	PI*R/T	sinR/R
2	pt pris comme centre y=0 x=0	296511,2303911	2341309,0379671	3,40829E-005	1,0707E-009	100,00
3	0	313941,4400035	2174763,9094206	167454,74631	5,260746008	-16,22
4	1	289099,3765517	2217575,9225980	123954,90884	3,89415831	-17,55
5	2	410406,7177238	2351413,8103081	114342,85485	3,592186728	-12,12
6	3	331285,1779878	2381739,9734885	53328,11618	1,67535218	59,36
7	4	202405,3924327	2314761,7986652	97778,65133	3,071806927	2,27
8	5	118503,2069703	2409566,5974380	190646,14034	5,989325139	-4,84
9	6	220675,6121469	2408168,7508539	101100,25815	3,176158283	-1,09

Extrait du tableau de données fichier.dbf du shp (coord. projection Lambert2)

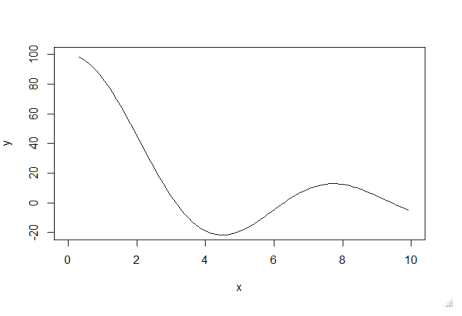
RQ. Les données se trouvent dans les fichiers :
`Donnees/Donnees_Chapitre_3/Test-krigeage-random-points-et-sinR-R.csv`

et *`Donnees/Donnees_Chapitre_3/Test-krigeage-random-points-et-sinR-R.ods`*

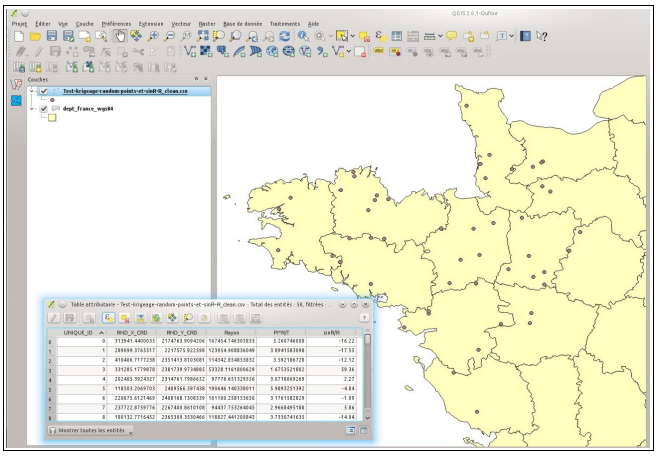
RQ. Les valeurs calculées pour les rayons étant de l'ordre de la centaine de milliers de mètres (colonne D), il a été nécessaire de diviser le résultat par 100000 (colonne E du tableau ci-dessous) pour obtenir des valeurs F(R) *avec une amplitude évoluant entre +100 (au point d'origine) et -20 (colonne F). On note que les valeurs des rayons sont comprises entre 0 et 2 (après division par 100000), soit des valeurs de 0 à 6,28 après multiplication par Pi (colonne E) pour obtenir des valeurs en radians. Voir fichier 'ods' pour le détail des calculs.*

```
# domaine des variations de la fonction (valeur 0 exceptée)
x = 0.1*pi : 100;
y = 100*sin(x)/ (x) ;
# On trace ensuite la courbe y = f(x)
plot (x, y, 'l', xlim=c(0,10), ylim=c(-20,100))
```

*Graphique correspondant à la fonction F(x) = 100 Sin (x) / x ; avec x = (R/100000)*Pi*

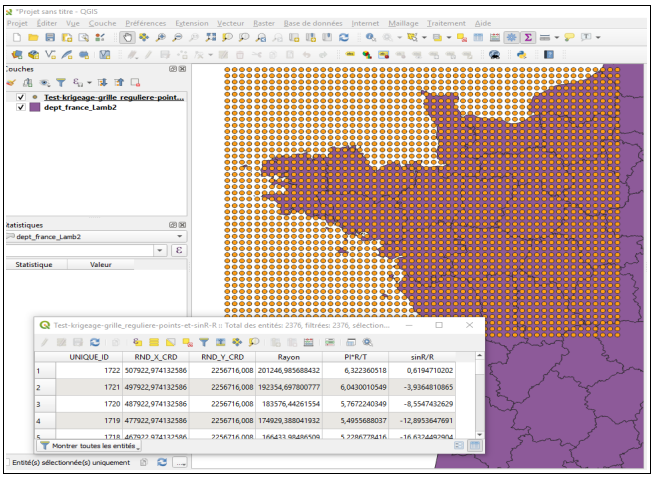


Utilisez un logiciel SIG pour visualiser les données (les coordonnées sont en Lambert2 : EPSG : 27572). Sur Qgis on utilise la commande : couche -> ajouter une couche de texte délimité. Indiquer le “;” comme délimitateur de champ et la “,” en séparateur décimal.



Visualisation du jeu de données sous Qgis

En fonction du temps, on pourra aussi générer un jeu de données réparties sous la forme d'une grille régulière.



Pour générer cette grille, utilisez l'onglet "Vecteur" -> outil de Recherche -> Point-réguliers. Il faut ensuite créer les champs RND_X_CRD ; RND_Y_CRD à l'aide de la calculatrice de la table attributaire (en mode édition). Créez des champs de type nombre decimal X et Y et indiquez dans le champs de calcultarice directement les coordonnées courantes $X et $Y. Sauvez ensuite la grille sous format shape, puis travaillez sur le fichier tableur pour calculer les champs R et F(R) = 100*Sin(R)/(R). Enregistrez le fichier en csv avec comme séparateur le “;”.

3.1 Méthode d'interpolation des Splines

– Principe :

Cette méthode d'interpolation est basée sur le **principe des régressions polynomiales.** Une spline est une fonction définie par morceaux et par des polynômes.

Les points (X,Y) à partir desquels sont calculés les valeurs interpolées correspondent à des relevés géographiques (ex. pt de températures).

Le cas le plus courant des splines est la spline « **cubique** ». Elle est uniforme et définie par des **polynômes de degré 3.**

– Exemple :

Interpolation par la méthode des Splines sur un jeu de données dont les valeurs z sont données par la formule $F(R) = 100 \sin R / R$ avec $R=(x^2 + Y^2)^{1/2}$

Exemple de script R avec génération de l'interpolation par les splines

Utilisation de Interp() du package AKIMA.

interp(x, y, z, xo=seq(min(x), max(x), length = 40), yo=seq(min(y), max(y), length = 40), linear = TRUE, extrap=FALSE);

avec :

x vecteur des coordonnées en x. Pas de valeurs manquantes

y vecteur des coordonnées en y. Pas de valeurs manquantes

z vecteur des valeurs du paramètres en x et y. Pas de valeurs manquantes .

x, y, et z de même taille et n>4

xo vecteur des coordonnées en x de la grille. 40 points par défaut. parametrable xo=seq(min(x), max(x), length = 100)

yo vecteur des coordonnées en y de la grille

linear logical : indique si l'interpolation est linéaire ou par spline.

extrap logical : indique si l'interpolation se fait à l'exterieur du domaine de x et y?

exemple de Script R :

?read.csv

chargement du fichier.csv de points et valeurs

```
data<-read.csv( file="/Donnees/Donnees_Chapitre_3/Test-krigeage-random-points-et-sinR-R.csv",header=TRUE,sep=";",dec=",");
```

génération d'une image par interpolation de type spline et extrapolation

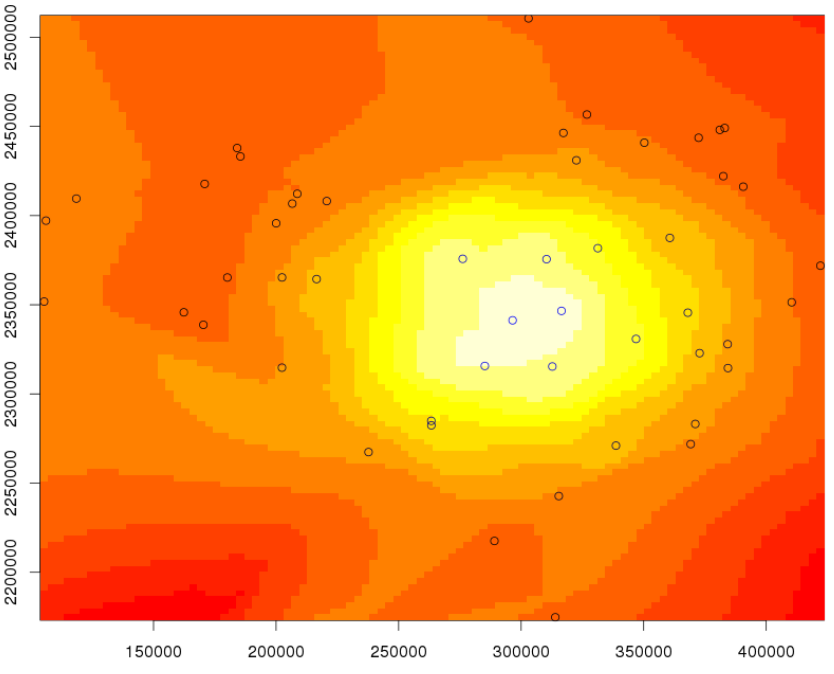
parametrage de xo et yo pour jouer sur le nb de cases

utiliser contour() pour afficher les résultats sous la forme de lignes de niveaux puis image() pour afficher une grille.

```
image(interp(data[,2], data[,3], data[,6], linear=FALSE, extrap=TRUE, xo=seq(min(data[,2]), max(data[,2]), length = 100), yo=seq(min(data[,3]), max(data[,3]), length = 100)));
```

#ajout des points d'origine sur la grille ou les lignes de niveaux

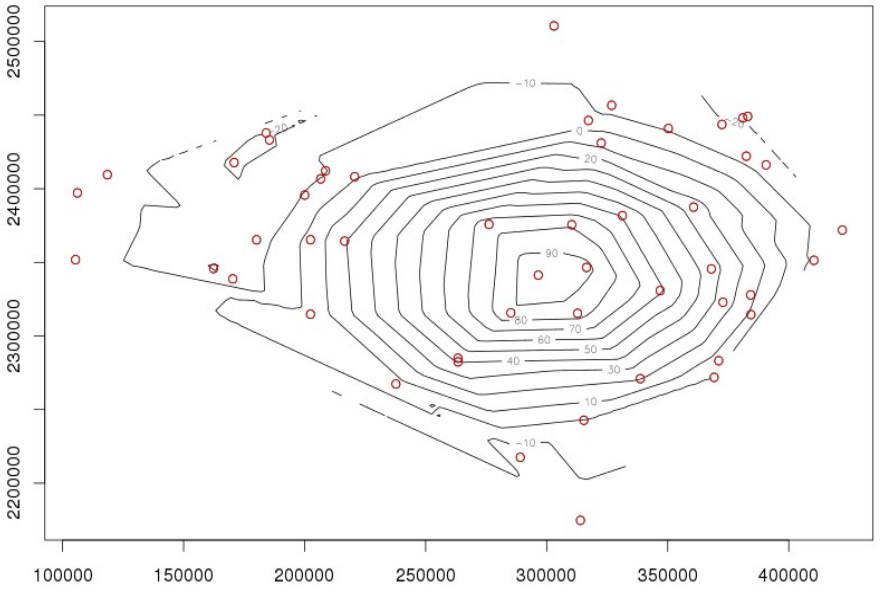
```
points(data[,2], data[,3]);
```



Exemple de résultat avec la méthode des splines (cubique) avec extrapolation

génération des lignes de niveaux par interpolation linéaire (regression linéaire) et sans extrapolation avec contour()

```
contour(interp(data[,2], data[,3], data[,6], linear=FALSE,
extrap=TRUE, xo=seq(min(data[,2]), max(data[,2]), length =
100), yo=seq(min(data[,3]), max(data[,3]), length = 100)));
```



Exemple de résultat avec la méthode des splines (régression linéaire) sans extrapolation

– Exercices :

- Tester des interpolations avec une régression de type cubique puis linéaire.

- Tester les calculs sans extrapolation.

- Jouer sur les séquences et le nb de pas (length) permettant de définir la grille.

- Utiliser image() pour obtenir une grille, contour() pour obtenir des lignes de niveaux.

- Refaire les tests en retirant des points du nuage de points source (attention à faire une copie des points d'origine).

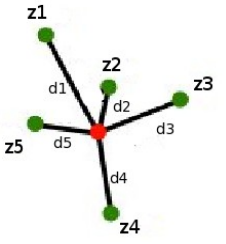
- Refaire les tests avec un jeu de données fourni sous la forme d'une grille régulière de points, puis en soustrayant des points à cette grille.

- Retrouve t-on la forme du "dôme" de la sinusoïde? Regardez les différences. Quel est l'impact sur l'interpolation de la présence ou l'absence de points ou de données fournies sous la forme de grilles régulières. A quoi sont dues les différences? L'interpolation résiste t elle à la soustraction de points?

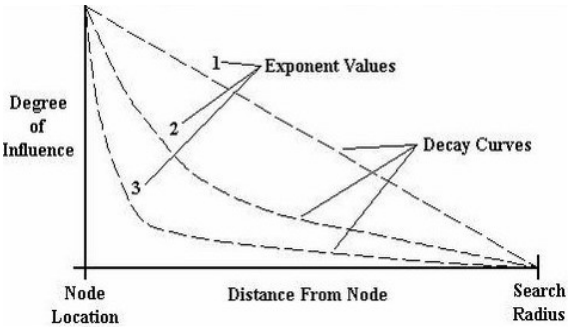
3.2 Méthode d'interpolation par pondération par l'inverse de la distance (IDW)

– Principe :

Cette méthode fait intervenir essentiellement les distances “di” qui séparent un pixel “s” donné des points d’échantillonnages. La valeur interpolée z(s) à la position “s” du pixel considéré peut être calculée à partir des N observations voisines, selon une formule (moyenne arithmétique pondérée).



Cette formule fait notamment intervenir une **exposant** α (valeur supérieure ou égale à 0, fixée par l’utilisateur). L’effet de la distance sur les valeurs interpolées sera d’autant plus faible que l’exposant α est fort.



Le cas $\alpha = 2$ correspond au modèle de pondération gravitaire le plus utilisé.

D’autre part, il faut fixer le nombre N de points voisins pris en compte dans le calcul de la moyenne. Le recrutement des points voisins dépend de cela et peut grandement influencer le résultat..

– Exemple :

Interpolation par la méthode IDW sur un jeu de données dont les valeurs z sont données par la formule **F(R) = 100 Sin R / R** avec $R=(x^2 + y^2)^{1/2}$

Exemple de script R avec génération de l'interpolation par IDW avec la librairie Gstat

chargement du fichier.csv de pts et valeurs

```
data<-read.csv(file="/Donnees/Donnees_Chapitre_3/Test-
krigeage-random-points-et-sinR-R.csv",
header=TRUE,sep=";",dec=",");
```

#construction du dataframe / tableau à 3 colonnes contenant les données à interpoler

```
Mesures<- data.frame(LON=data$RND_X_CRD, LAT=data$RND_Y_CRD, Z =
data[,6]);
```

#construction d'un objet "Mesure" avec la fonction "coordinates()" : Mesure devient un objet de "classe spatiale" qui contient les coordonnees des mesures à interpoler.

```
coordinates(Mesures)<-c("LON","LAT");
```

 # Estimation sur les noeuds d'une grille

#1. Construisez les séquences de coordonnées de la grille sur laquelle aura lieu l'estimation couvrant toute la zone avec un pas de 1000 :

```
MinX <- min(data[,2])
MinY <- min(data[,3])
MaxX <- max(data[,2])
MaxY <- max(data[,3])
```

on fabrique des séquences avec un pas de 1000mètres (soit 317 entités pour x et 336 pour y). On créé ainsi les coordonnées X et Y des points de la grille.

```
Seqx <- seq(MinX, MaxX, by=1000)
Seqy <- seq(MinY, MaxY, by=1000)
```

#2. Construisez la liste de tous les noeuds de cette grille : construction de la grille

rep() : function qui permet de répliquer la Seqx par le nombre de Seqy et inversement la Seqy par Seqx. On obtient ainsi deux matrices de valeurs respectivement de X et de Y (nombre d'entités équivalentes = 106512).

```
MSeqx <- rep(Seqx, length(Seqy))
MSeqy <- rep(Seqy, length(Seqx))
```

sort() permet de réaliser un tri sur la séquence des valeurs de MSeqy. Etape permettant de mettre en correspondance les valeurs de MSeqx et MSeqy afin de reconstituer les coordonnées d'une grille (utilisation de la function "data.frame()").

```
MSeqy <- sort(MSeqy, decreasing=F)
Grille<-data.frame(X=MSeqx, Y=MSeqy)
```

création d'un objet de la classe spatiale (utilisation de coordinates) et indication que l'objet est une grille (utilisation de gridded())

```
coordinates(Grille)=c("X","Y")
gridded(Grille)<-TRUE
```

visualisez l'objet de la classe spatiale

```
print(Grille)
```

#3. Construisez une estimation par IDW sur tous les noeuds de la grille (la #liste construite ci-dessus) :

idp l'exposant (ex. 2) pour le model gravitaire

Le premier paramètre (Z~1) est prévu pour insérer une formule dans le cadre plus large du krigeage (cf. function ?idw). On ne l'utilise pas pour l'idw.

location : objet de la "classe spatiale". Paramètre correspondant à l'objet qui va contenir les coordonnées spatiales des données à interpoler.

newdata : correspond aux coordonnées de la grille que l'on cherche à générer. Ce sont en fait les positions des noeuds de la grille.

maxdist : paramètre pour indiquer une distance maximum de recrutement des points voisins

RQ . faire "?idw" pour avoir plus de renseignements

```
PredictionIDW <- idw(Z~1, locations=Mesures,
newdata=Grille, idp = 2.0)
```

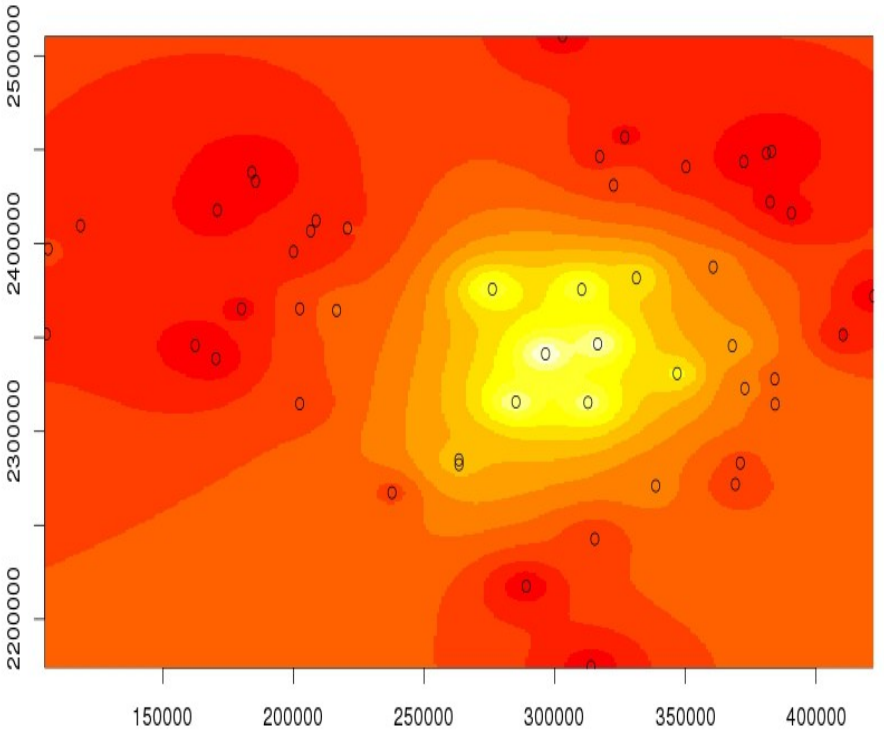
RQ . Les valeurs interpolées sont dans `PredictionIDW$var1.pred` et contient 106512 éléments (autant que la grille)

#4. Vous allez chercher à afficher le contenu de Prediction avec la fonction image. Celle-ci prend en paramètre un objet de type list contenant deux vecteurs de coordonnées géographiques de la grille nommés x et y, et une matrice de données spatialisées nommée z (formée des valeurs interpolées `PredictionIDW$var1.pred` soit 106512 éléments et length(Seqx) permettant surtout d'avoir un identifiant ou compteur) :

```
PredictionIDW.est <- list(x=Seqx, y=Seqy,
z=matrix((PredictionIDW$var1.pred), nrow=length(Seqx)))
```

#5. Dessiner la carte

```
image(PredictionIDW.est)

points(coordinates(Mesures))
```



Grille d'interpolation obtenue par la méthode de l'IDW (modèle gravitaire degré de pondération = 2)

– Exercices :

- Tester des interpolations en modifiant le degré de pondération (idp = 0 ou 1 ou 2 ou 3),

- Tester des interpolations en modifiant le "maxdist" (distance maximale pour recruter les points voisins)

ex. 5000, 10000, 25000, 50000, 100000, 150000 mètres.

- Tester des interpolations en modifiant le "nmax" (nb max de points voisins pris en compte) et le nmin (nombre minimum de points voisins)

ex. 3, 12, 25, 50 points.

exemple :

```
PredictionIDW <- idw(Z~1, locations=Mesures, newdata=Grille, idp = 2.0, nmax=25, nmin=3, maxdist=100000)
```

- Testez l'interpolation avec un jeu de données fourni sous la forme d'une grille régulière. Quelle est la qualité de l'interpolation?

- Refaire les tests en retirant des points des jeux de données source (attention à bien dupliquer les fichiers “sources”).

- Retrouve t-on la forme du "dôme" de la sinusoïde? Regardez les différences, l'impact sur l'interpolation de la présence ou l'absence de points. A quoi sont dues les différences?

3.3 Méthode d'interpolation par Krigeage

– Principe :

3.3.a. Réalisation d'un variogramme

Le variogramme permet de savoir dans quelle mesure « ce qui se passe en un point » ressemble ou non en moyenne à ce qui se passe dans son entourage pour une distance donnée.

Il met en relation des intervalles de distances (ou lag) avec des variances moyennes établies pour des valeurs de couples de points dont la distance appartient à un même intervalle de distance.

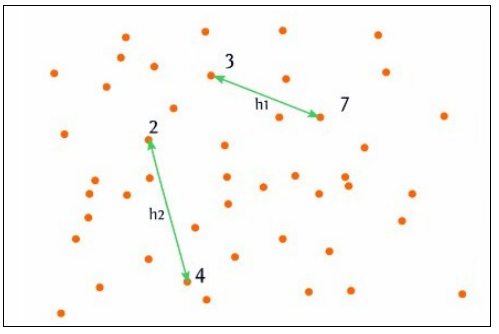


Figure montrant deux couples de points dont les distances appartiennent à un même intervalle "h" (ou lag). Il y a en réalité beaucoup plus que deux couples de points dans cet intervalle. Dans cet exemple, pour l'intervalle h, une variance moyenne sera calculée avec l'ensemble des variances de chaque couple de points.

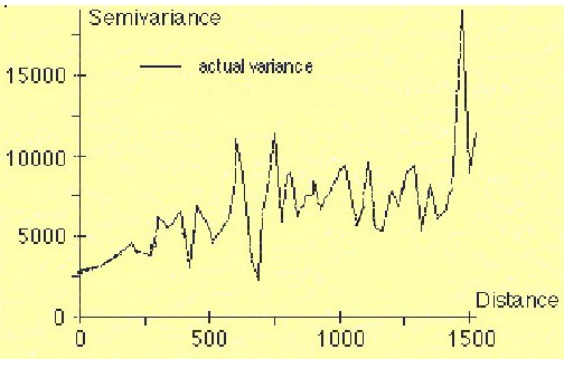
- Moyenne couple h1 = (3+7)/2 =5
- Variance couple h1 = $((3-5)^2+(7-5)^2)$ /2 = 4
- Moyenne couple h2 = (2+4)/2 =3
- Variance couple h2 = $((2-3)^2+(4-3)^2)$ /2 = 1
- Variance couple h3 ...

On calcule ensuite pour chaque intervalle **la moyenne** des variances.

Exemple pour l'intervalle "**h**", si **h1** et **h2** appartiennent à cet intervalle h :

Moy. variance interv. h = (Var. couple h1 + Var. couple h2 +..) /nb couples de h

Rq. Lors du calcul des moyennes, les calculs pour chaque couple de points sont en fait réalisés deux fois car le calcul est lancé point après point vis à vis de l'ensemble des autres points. Lors du calcul de la moyenne des variances pour un intervalle il est donc possible de diviser par deux le nombre de couples de points pris en compte d'où le terme de **semi-variance** souvent rencontré.



Exemple de semi-variogramme mettant en relation des intervalles de distance en abscisses et des semi-variances en ordonnées

En théorie, la variance de la variable z(s) diminue lorsque les points se rapprochent jusqu'à atteindre une variance nulle.

Rq : En théorie, lorsque le nombre de points de valeurs est élevé (à partir d'un millier de points), le lag interval correspond à la distance minimale obtenue parmi l'ensemble des paires de points. Néanmoins, le nombre de points n'est pas toujours très élevé (cas de 50-100 maximum). Aussi, il est nécessaire de recruter davantage de paires de points pour obtenir des « moyennes de variances représentatives » pour chaque intervalle. D'où la nécessité de rechercher des « lag interval » adaptés à chaque ensemble de points pour chaque cas de figure.

Il est donc conseillé de vérifier le nombre de couples de points recrutés pour chaque intervalle afin de choisir l'intervalle du variogramme. 30 couples de points par intervalle semble un minimum pour établir une moyenne de variance représentative.

3.3.b. Modélisation à partir du variogramme (ou semi-variogramme) : courbe d'interpolation

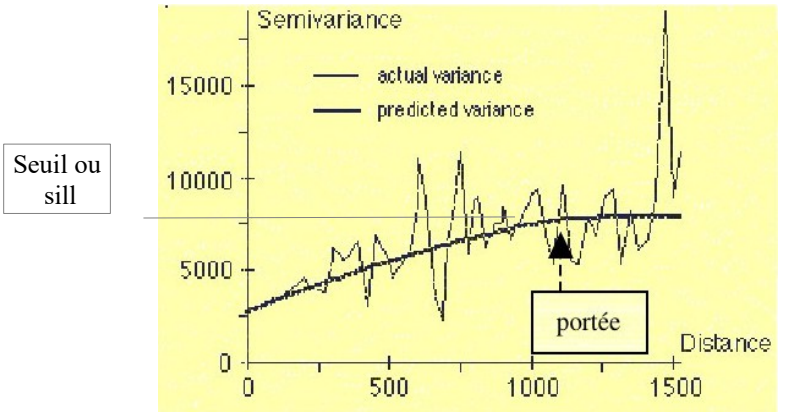
Une fois le semi-variogramme réalisé, il est ensuite nécessaire de calculer une « courbe d'interpolation » ou modèle traduisant le mieux possible la tendance du semi-variogramme.

Les logiciels proposent généralement plusieurs types de modèles : circulaires, sphériques, exponentielles, linéaires avec plusieurs degrés.

La portée traduit la distance à partir de laquelle il n'y a plus de relation "apparente" de dépendance entre distance et variance.

Le seuil (Sill) est à mettre en rapport avec le plateau apparent des valeurs de variances obtenues à partir d'une certaine distance.

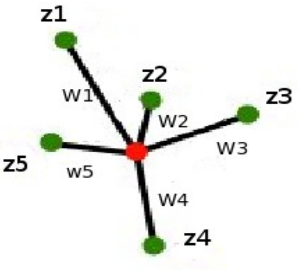
Il n'est pas toujours possible d'obtenir une courbe de tendance. La regression donne parfois une ligne horizontale (portée dès h=0), cela traduit une répartition de la donnée homogène pour l'échelle et le lag (intervalle) choisis.



Exemple : Semi-variogramme et sa courbe de prédiction ou modèle.

3.3.c. Calcul d'une grille d'interpolation (raster) dont les valeurs sont calculées sur la base du variogramme.

Cette étape se concrétise par le calcul, pour chaque pixel de la grille, d'une moyenne des valeurs pondérées d'un ensemble de N points voisins. (effectif paramétrable) et cela sur un rayon qu'il est généralement possible de fixer. Les valeurs de chaque point voisin sont pondérées par un coefficient Wi qui est fonction de la courbe du variogramme.



Les poids Wi sont fonction du degré de similarité entre les points, c'est-à-dire de la covariance entre les points en fonction de la distance qui les sépare.

Notons que les krigeages des pixels qui sont éloignés des points d'études par des distances supérieures à la portée de la courbe d'interpolation, ne sont en fait alors que des moyennes des valeurs des points d'études, car le coefficient Wi qui est fonction du variogramme est alors le même pour tous les points pris en compte dans le calcul de la moyenne.

RQ : Le choix de la courbe du semi-variogramme est effectué après avoir calculé les semi-variogrammes correspondant à plusieurs « lag interval », le semi-variogramme apparaissant le plus représentatif de l'ensemble des semi-variogrammes calculés est alors retenu. Par ailleurs, on peut se référer à l'indice « RMS error » fourni lors du calcul de chaque semi-variogramme et courbe d'interpolation pour appuyer le choix du semi-variogramme et le type de courbe de régression polynomiale.

Le « Root Mean Square (RMS) error » représente la différence entre les valeurs originales et les valeurs obtenues par le calcul de la courbe de régression polynomiale. Cet indice permet donc de mesurer le degré d'adéquation entre le semi-variogramme et la courbe de régression polynomiale. Plus les valeurs de « RMS error » sont basses et plus la surface interpolée représente bien les données de chaque point.

– Exemple :

Interpolation par la méthode du krigeage sur un jeu de données dont les valeurs z sont données par la formule **F(R) = 100 Sin R / R** avec $R=(x^2 + Y^2)^{1/2}$

Exemple de script R avec génération de l'interpolation par krigeage avec la librairie Gstat

chargement du fichier.csv de pts et valeurs

```
data<- read.csv(file="/Donnees/Donnees_Chapitre_3/Test-
krigeage-random-points-et-sinR-
R.csv",header=TRUE,sep=";",dec=",");
```

#1. Construction de l'objet spatial "Mesures" avec la fonction coordinates()

#construction dataframe / tab à 3colonnes contenant les mesures :

```
Mesures<- data.frame(LON=data$RND_X_CRD,
LAT=data$RND_Y_CRD, Z = data[,6]);
```

#à ce niveau Mesures n'est pas encore un objet spatial

Précisez quelles colonnes contiennent les coordonnées géographiques :

```
coordinates(Mesures)<-c("LON", "LAT");
```

Mesures devient un objet spatial avec deux colonnes pour coordonnées

#2. Construisez et examinez le variogramme expérimental avec la fonction variogram() :

cutoff=200000 : estimation de la portée (choisir une distance englobant la portée, donc un peu plus longue que la portée)

Z~1 : valeur par défaut. Ce paramètre est saisi lors de l'étape suivante avec la courbe d'interpolation.

Mesures : objet de classe spatiale contenant les points de mesures avec les coordonnées et valeurs

```
VariZ <- variogram(Z~1, Mesures, cutoff=200000);
```

pour jouer sur les lag (intervalles) on indique le paramètre "width"

```
VariZ <- variogram(Z~1, data=Mesures, width=25000,
cutoff=200000);
```

On choisit un lag de l'ordre de 25Km. Il faut s'assurer qu'il y a assez de paires de points dans chaque lag.

vérification du nombre de couples de points recrutés pour chaque intervalle avec un pas de 25000 mètres :

```
VariZ
```

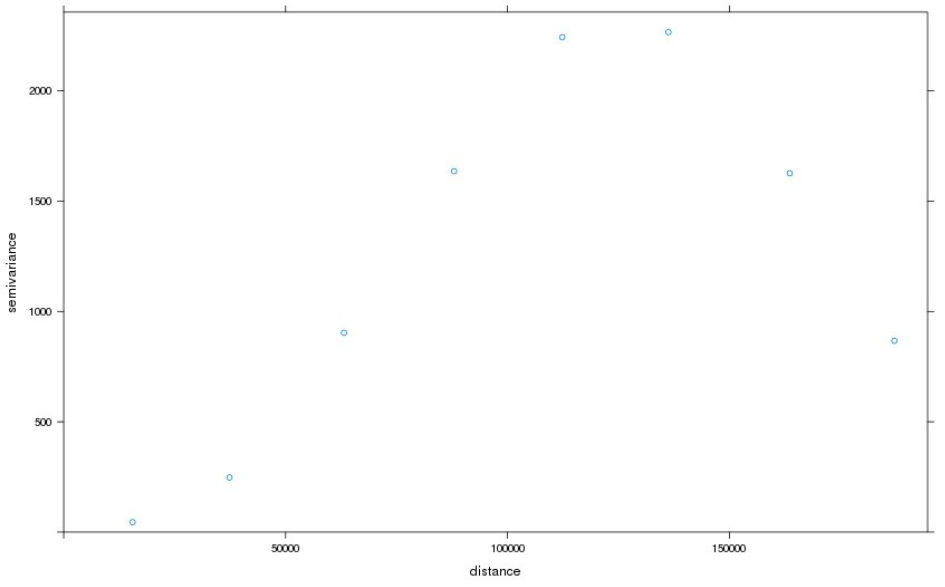
```
   np      dist      gamma dir.hor dir.ver   id
1  31  15512.91   44.83768       0       0 var1
2  81  37349.61  247.49299       0       0 var1
3 128  63141.69  903.64523       0       0 var1
4 137  87973.08 1635.94016       0       0 var1
5 165 112324.01 2243.48877       0       0 var1
6 148 136259.20 2266.92983       0       0 var1
7 129 163597.65 1626.69972       0       0 var1
8 144 187173.50  867.18974       0       0 var1
```

Rq. Faire plusieurs tests! Attention, ici il faut aller à 200000 et non seulement à 100000 mètres!

Exercice :

- Tester plusieurs intervalles, plusieurs portées et regarder le nombre de couples recrutés.

On réalise ensuite une représentation du semi-variogramme
`plot(VariZ)`



Semi-variogramme avec un lag interval de 25Km et un cutoff de 200000

Rq. La fonction variogram() fournit directement des semi-variances.

Rq. La représentation des petites distances est un vrai défi car il y a souvent des variances inexpliquées.

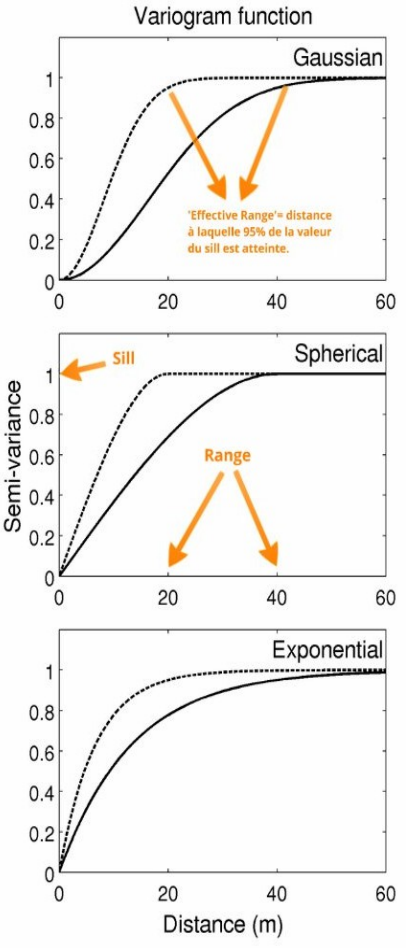
Rq. A cette étape le variogramme peut être déjà intéressant à étudier! L'identification de la portée permet de définir la distance maximale sur laquelle va s'exercer une relation de dépendance entre les valeurs.

Rq. La forme du variogramme montre comment la relation de dépendance entre les valeurs évolue en fonction de la distance qui les sépare.

#3. Ajustez un modèle de variogramme

Cette étape va consister à paramétrer la courbe ou modele de variogramme. Le parametrage du modèle de variogramme se fait avec la function **vgm()** et notamment les 3 paramètres "psill", "model" et"range".

Plusieurs functions ou "type de modèle" sont à disposition : courbe de Gauss, Spherique, Exponentielle : cf. forme sur les trois graphiques:



Choisir le model qui répond le mieux à la forme du nuage de points du variogramme.

En ce qui concerne l'exemple précédent : modèle de Gauss parait le mieux correspondre.

Rappel : "Sph" : model spherique; "Lin" : linéaire; "Gau": Gausse; "Exp": Exponentiel ; "Log" : Logarithmique.

Voir : `>show.vgms()` pour avoir l'ensemble des modèles.

Le choix du seuil (Sill) est à mettre en rapport avec le plateau apparent.

Le range correspond à la distance où le plateau (ou portée) est atteinte. (Rq. en fonction du modele choisi, psill et range ne sont pas toujours utiles, ils sont calculés dynamiquement, cf. `> str(VariZFitted))`

Dans l'exemple précédent (variogramme au lag de 25Km), on peut estimer le plateau vers 2500 (voir 3000).

Le range, distance à laquelle 95% du seuil est atteint est de l'ordre de 150000m (150 Km);

Rq. Après le range et surtout la portée, la baisse de la variance à moins de sens.

Exemple :

```
VariZFitted = fit.variogram(VariZ, vgm(psill=2500,
model="Gau", range=150000));
```

Le nombre de combinaisons possibles pour paramétrer le modèle devenant important (choix du "psill", "model" et"range"), il est utile de regarder l'indicateur "SSErr" afin de

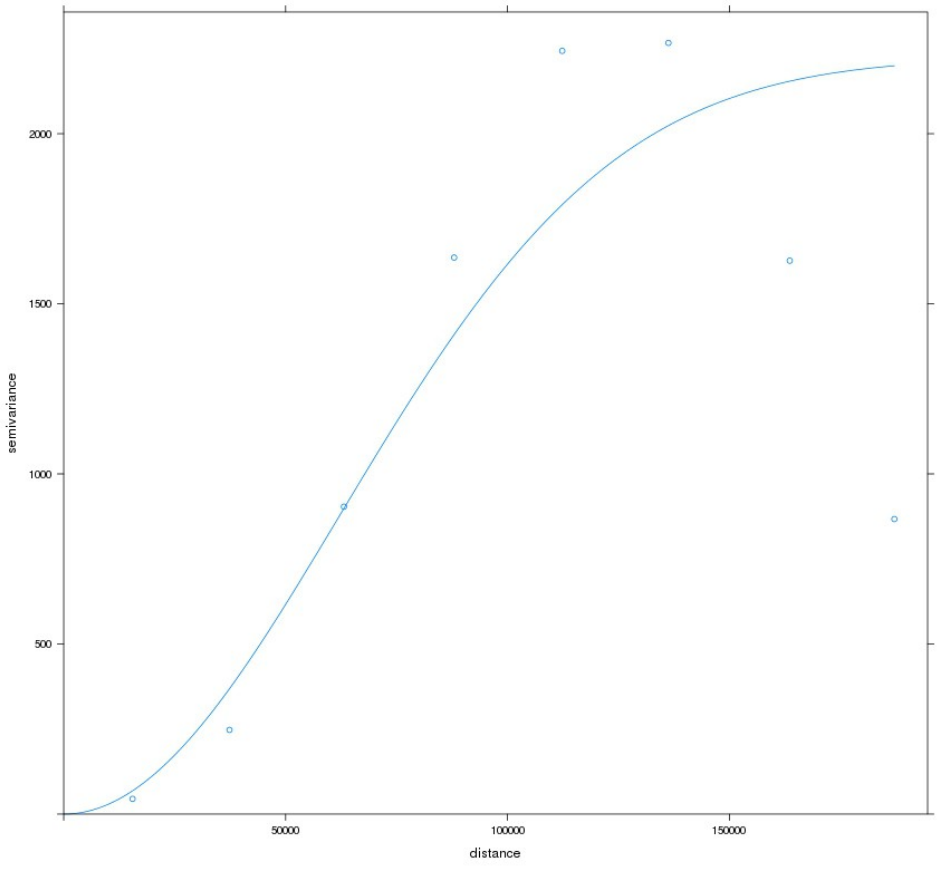
choisir le modèle. Il faut aussi regarder la forme du modèle par rapport à la forme du variogramme.
Pour cela faire **str(**VariZFitted**)** pour avoir cet indicateur d'erreur. Le SSErr est un indicateur de l'ajustement du model au variogramme. Il correspond à la somme des erreurs au carré du modèle ajusté. Plus il est faible mieux c'est. (cf. explications du RMS en début de chapitre dans les principes et aussi du coefficient de détermination,).

```
> str(VariZFitted)
Classes ‘variogramModel’ and 'data.frame':      1 obs. of
9 variables:
 $ model: Factor w/ 20 levels "Nug","Exp","Sph",..: 4
 $ psill: num 2223
 $ range: num 87748
 $ kappa: num 0.5
 $ ang1 : num 0
 $ ang2 : num 0
 $ ang3 : num 0
 $ anis1: num 1
 $ anis2: num 1
 - attr(*, "singular")= logi FALSE
 - attr(*, "SSErr")= num 0.0136
```

ex. SSErr pour le modele de Gauss = 0,0136; pour le modele Sph = 0,0273

#4. Représentez graphiquement le variogramme expérimental et son modèle ajusté

```
plot(VariZ, VariZFitted);
```



Modele de Gauss dans le (semi-)variogramme

#Estimation sur les noeuds d'une grille

#1. Construisez les séquences de coordonnées de la grille sur laquelle aura lieu l'estimation couvrant toute la zone avec un pas de 1000 (cellules de 1x1km):

```
MinX <- min(data[,2])
MinY <- min(data[,3])
MaxX <- max(data[,2])
MaxY <- max(data[,3])
Seqx <- seq(MinX, MaxX, by=1000)
Seqy <- seq(MinY, MaxY, by=1000)
```

#2. Construisez la liste de tous les noeuds de cette grille : construction de la grille

```
MSeqx <- rep(Seqx, length(Seqy))
MSeqy <- rep(Seqy, length(Seqx))
MSeqy <- sort(MSeqy, decreasing=F)
Grille<-data.frame(X=MSeqx, Y=MSeqy)
coordinates(Grille)=c("X","Y")
gridded(Grille)<-TRUE
```

#3. Construisez une estimation par krigeage sur tous les noeuds de la grille (cf. la liste construite ci-dessus). Pour cela utilisons la fonction **krige()** permettant de calculer les valeurs Z sur les noeuds de la grille à partir des Mesures et du modèle du variogramme (VarizFitted)

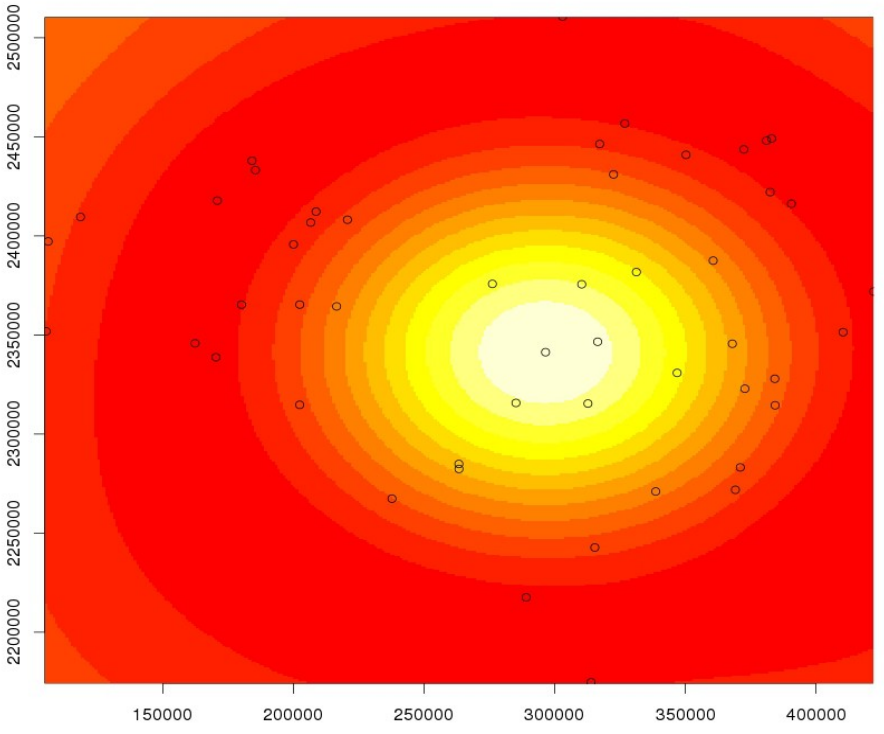
```
PredictionKrige <- krige(Z~1, locations=Mesures, newdata=Grille, VariZFitted)
```

#4. Vous allez chercher à afficher le contenu de PredictionKrige avec la fonction **image()** préalablement employée. Celle-ci prend en paramètre un objet de type list contenant deux vecteurs de coordonnées géographiques nommés x et y, et une matrice de données spatialisées nommée z (contenant les valeurs estimées dans predictionKrige) :

```
PredictionKrige.est <- list(x=Seqx, y=Seqy, z=matrix((PredictionKrige$var1.pred), nrow=length(Seqx)))
```

#5. Dessiner la carte

```
image(PredictionKrige.est)
points(coordinates(Mesures))
```



Grille d'interpolation obtenue par krigeage avec un lag de 25km, un modèle de Gauss

– Exercices :

1. Faire différents tests de Variogrammes avec :

- différent intervalles (ex. 1000, 10000, 25000, 50000, 100000 mètres)
- en modifiant la portée (ex. 25000, 50000, 100000, 200000, 300000 mètres)
- afficher les graphiques

2. Construire les courbes prédicitives :

- en testant les différents modèles à disposition (Linéaire, Gausse, Sphérique, Exponentielle)
- en modifiant les valeurs du seuil, du range
- calculer le SSerr
- afficher les graphiques

3. Construire les grilles interpolées

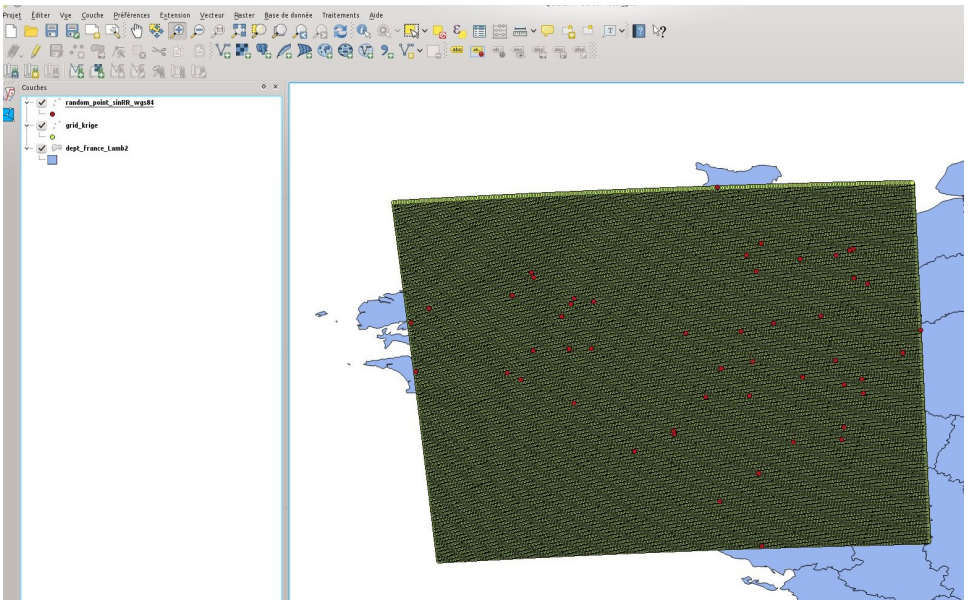
- Retrouve t-on la forme du "dôme" de la sinusoïde
- Discuter les valeurs interpolées, les valeurs extrapolées,
- Quel est l'impact de la proximité d'un point vis à vis des valeurs interpolées. As t on un effet de lissage, ou au contraire les points ont ils des effets locaux?

4. Rejouer les procédures en retirant des points.

Regarder l'impact sur les résultats.

#6. utilisation de interp2xyz pour générer un tableau csv avec x,y,z (importable sur QGis)

```
write.csv(interp2xyz(predictionKrige.est),
file="/home/lgaudin/z_mnt/projet/6_FORMATION/CONTENU/19_ini
tiation_geostat/donnees/grid_krige.csv");
```



#7. sortie d'une grille (asc) : utilisation de `write.asciigrid {sp}`

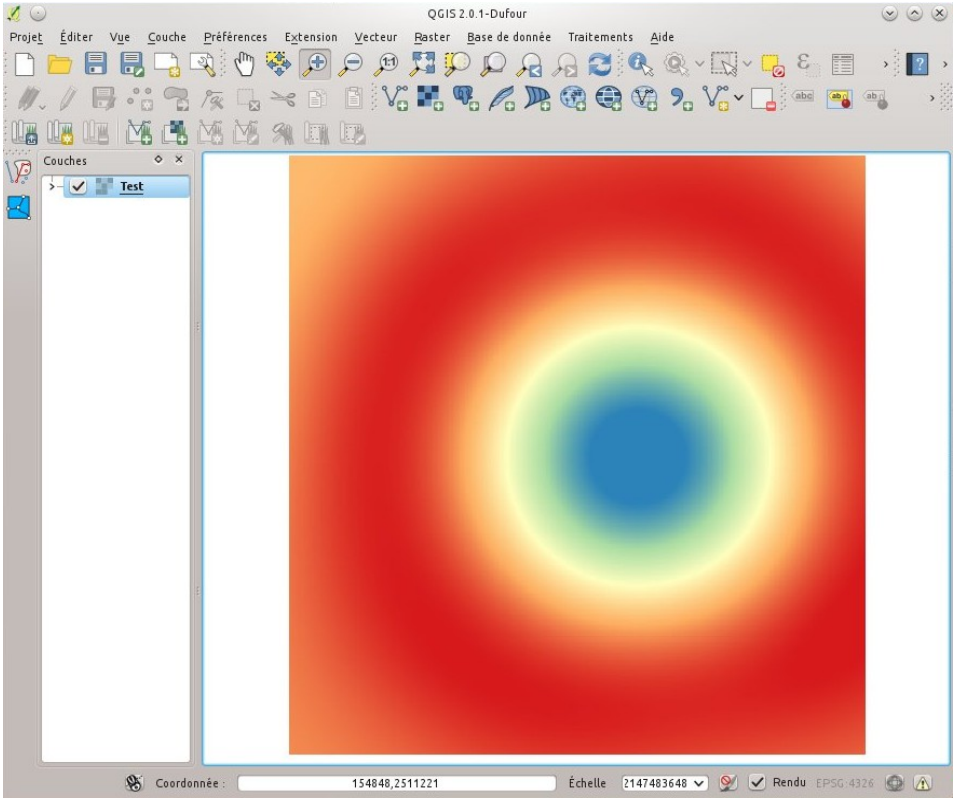
Pour éviter les problèmes d'arrondis sur la taille de cellules de la grille en sortie , car parfois il existe des différences infimes entre la taille des cellules, on force les dimensions des cellules à la moyenne des côtés. On récupère les 2 valeurs de côtés dans la liste de 2 éléments cs. (cs est un couple de valeurs)

```
cs <- slot(slot(PredictionKrige, "grid"), "cellsize")
```

On remplace ensuite les valeurs existantes par la moyenne des 2 côtés.

```
slot(slot(PredictionKrige,    "grid"),    "cellsize")    <-
rep(mean(cs), 2)

write.asciigrid(PredictionKrige,
"/home/lgaudin/z_mnt/projet/6_FORMATION/CONTENU/19_initiati
on_geostat/donnees/Test.asc")
```

On peut ensuite ouvrir cette grille de type asc dans un sig, type Qgis.



Exemple : grille issue de l'interpolation ouverte sur QGis

Chapitre 4 : Recherche de corrélations entre des distributions spatiales de plusieurs paramètres.

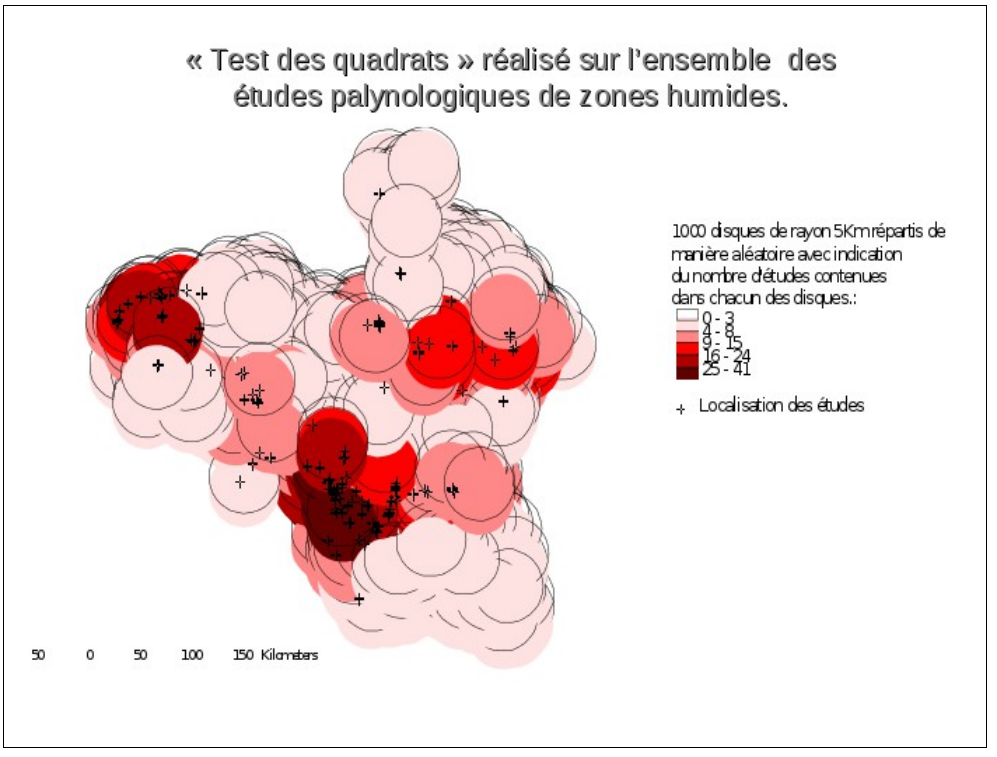
La mise au point de protocoles permettant d'étudier les distributions spatiales de données nécessite la création et manipulation de couches géographiques. Ces travaux sont alors grandement facilités par l'utilisation de logiciels de Systèmes d'Informations Géographiques (ex. QGis). Ces logiciels proposent généralement des outils de recherche, de géotraitement, de géométrie nécessaires à la construction de ces protocoles.

Ce chapitre a été conçu sous forme d'exercices. Il est largement ouvert aux discussions et problématiques d'ordre spatiale que pourront apporter les personnes suivant la formation.

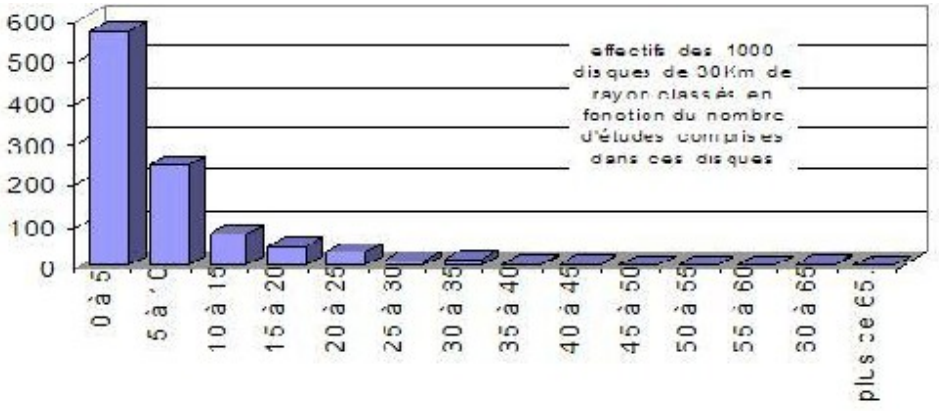
Exemple 1 : Etude de la distribution spatiale de points : Le test des Quadrats.

Le but de ce test est d'estimer la distribution spatiale de points : aléatoire, régulière, en agrégat. Il est nécessaire de juger de cette distribution pour pouvoir comparer les distributions spatiales de données entre paramètres.

Dans cet exemple, afin de juger du type de distribution d'observations paléoenvironnementales (croix noires) à la surface du Massif armoricain, 1000 disques de 5km de rayon ont été générés de façon aléatoire.



Pour chacun de ces disques, le nombre de croix intersectées a ensuite été attribué.



Pour identifier le type de distribution il faut ensuite calculer la variance et la moyenne du nombre d'intersections pour ces 1000 disques.

Dans le cas d'une distribution régulière, le nombre de croix intersecté sera globalement le même pour chaque disque. La variance sera donc faible.

Moyenne > Variance

Dans le cas d'une distribution en agrégats, certains disques montreront un nombre d'intersections important et d'autres aucune intersection d'où une forte variance.

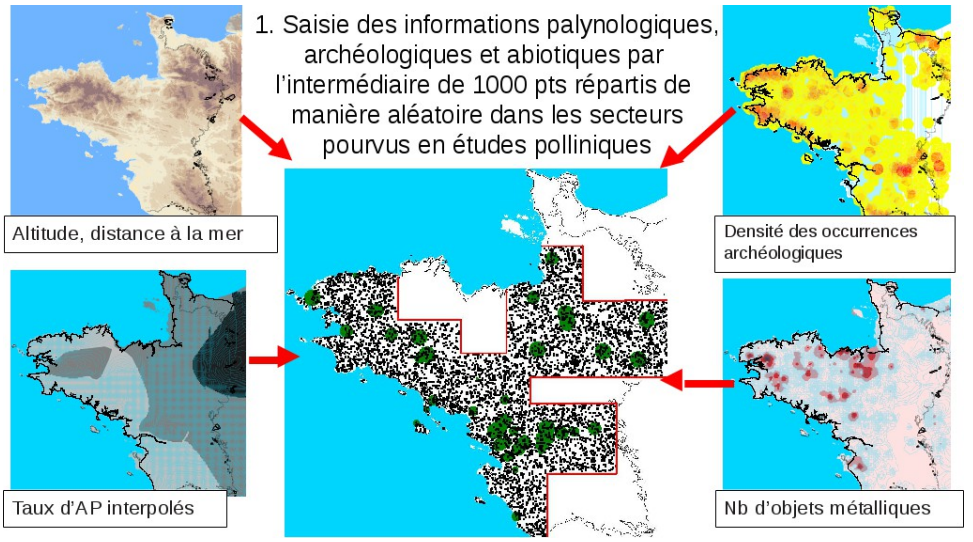
Variance > Moyenne

Dans le cas d'une distribution aléatoire, le nombre d'intersections est réparti sur l'ensemble des disques :

Moyenne # Variance

Exemple 2 : Recherche de corrélations entre les distributions spatiales de plusieurs paramètres.

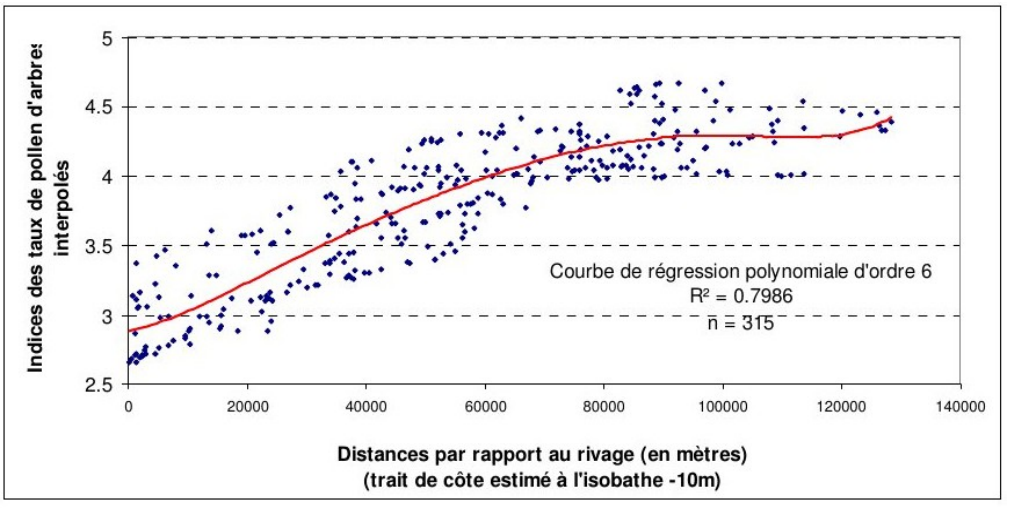
Exemple d'une recherche entre des distributions spatiales de paléo-forêts (taux de pollens d'arbres interpolés), de sites archéologiques, de données abiotiques (distances à la mer, altitudes, etc...).



Dans cet exemple, 1000 points ont été générés de façon aléatoire dans les zones se trouvant à proximité de zondages palynologiques (points verts). Pour chacun de ces points, une information de chacun des paramètres (paléo-forêts, archéologiques, abiotique) a ensuite été attribuée.

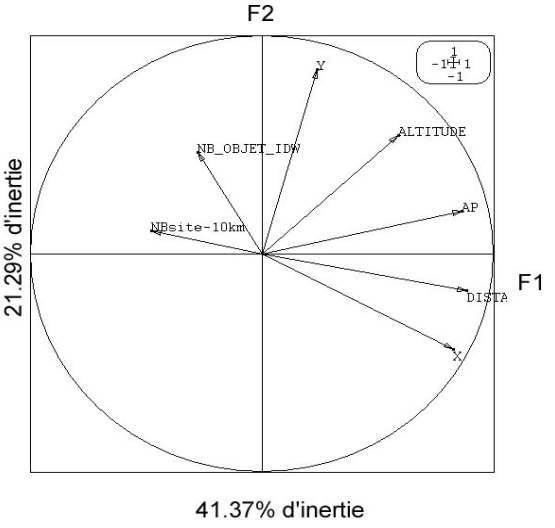
Ces 1000 points, “multiparamétrés”, peuvent ensuite faire l'objet d'une analyse en comparant les paramètres deux à deux (exemple avec le graphique ci-dessous sur l'effet de la continentalité sur la distribution des taux de pollens d'arbres interpolés), ou bien à l'aide d'analyses multivariées (exemple avec une ACP si l'ensemble des données sont de types quantitatifs).

Exemple de graphique mettant en relation la distance à la mer et les taux de pollens d'arbres interpolés à l'aide des 1000 points :



Une courbe de régression polynomiale a été générée.

Exemple d'Analyse en Composantes Principales réalisée sur les "1000 points multiparamétrés".



Constats :

Le taux de pollen d'arbres (AP) et Distance (Dista) à la mer sont corrélés.

En revanche densité archéologique et taux d'AP sont inversement corrélés.

Absence ou faibles corrélations avec les autres paramètres.

Chapitre 5 : Suggestions d'exercices.

Exercice 1 : Etude géostatistique du paléoenvironnement du Massif armoricain à l'aide de données de l'Age du Bronze

A partir des données sous format shp sur QGis:

- prendre la projection Lambert2 (EPSG : 27572)

- des valeurs ponctuelles de taux de pollens d'arbres de l'Age du bronze : information sur les paléo-forêts (fichier : *siteph-ab.shp*)

- des valeurs d'altitudes (Modèle Numérique de Terrain à 90x90m : *MNT_massifzoom*)

- des valeurs de distances à la mer (à calculer à l'aide du trait littoral) (fichier : *ligCLC.shp*)

- des valeurs de densités archéologiques de l'Age du Bronze

(fichiers : ponctuel : *bdsitearcheoabcorrige_wgs84.shp* grilles déjà calculées *: nbsiteab-10km.shp* et *nbobjet-idw.shp*)

1. Préparation des données : génération de 1000 points multiparamétrés

1.1 Faites un test des quadrats sur la distribution des analyses polliniques (taux d'AP) de l'Age du Bronze

- Faire des tests avec des disques de 5km, 10km, 50km
- Modifier la fenêtre

Rq. Utilisez les départements 22, 29, 56, 35 et 44 comme surface pour établir le nuage de points. (100 pt par département).Utilisez vecteur-> points aléatoires. Sélectionnez préalablement les départements de la couche dept_france_Lamb2.shp avant de lancer l'outil de génération de points aléatoires. Puis créez des zone tampons pour faire des disques puis faire l'intersection pour le comptage.(cf. Outils).

1.2 Faites des cartes d'interpolations des taux de pollens d'arbres (AB_AP) et de Poacées (AB_PO) de l'Age du Bronze à l'aide des différentes méthodes

- Splines (cubiques – sans extrapolation), IDW (modèle gravitaire), Krigeage (testez différents paramètres pour obtenir la meilleure interpolation. Utilisez les champs AB_AP, AB_PO sur le fichier *siteph-ab.shp*

Rq. Afin de pouvoir transformer par la suite le raster en vecteur, il faut prévoir une amplitude suffisante dans les valeurs de la grille interpolée. Il est donc nécessaire de x100 les valeurs d'AP du fichier d'origine afin d'avoir une colonne avec des valeurs d'AP allant de 100 à 500 et à terme d'obtenir une grille avec de telles valeurs.La transformation du raster en vecteur ne conserve que des entiers.

Rq. En fonction du temps restant ne tester que sur le paramètre "AB_AP" du fichier siteph-ab.shp

Rq. Lors du test sur les splines, tester uniquement l'interpolation en mode linéaire ou cubique (Linear = TRUE ou FALSE).

Rq. Lors du test en IDW : jouez sur les parametres "idp" (ou coef de ponderation de distance), "maxdist" (distance de recrutement des pts voisins), nmin et nmax pour le nb de pt voisins pris en compte : ex. #PredictionIDW <- idw(Z~1, locations=Mesures, newdata=Grille, idp = 3.0, maxdist=50000, nmin=10, nmax=15)

Rq. Lors du krigeage : pour établir le variogramme : paramétrez l'intervalle à prendre en compte pour recruter les couples de points (testez les intervalles de 5000 à 50000 mètres). La portée est de l'ordre de 200 à 300 km. Affichez les résultats avec la fonction plot(). Puis paramétrez la courbe de régression. Utilisez un psill (seuil de variance) de l'ordre de 15000 à 30000 et un range de 200000 à 300000). Regardez l'indicateur SSErr pour vous donner une idée de l'ajustement. Testez différents modèles d'ajustement (cf. Show.vgms()) pour avoir les différents modèles possibles.

Ex. de paramètres : Cutoff=200000 à 300000; width = 25000 à 50000; psill = 15000 à 30000; model = Exp ou Exc ou Lin ou Gau ou Cir etc...

Ex. Meilleur SSErr obtenu (0,0433) avec : CutOff = 200000; lag = 50000; Psill = 15000 ; Modèle = Exp ;

A noter que le modèle "Exc" semble étendre la relation distance – variance sur de plus longues distances.

- Des effets locaux sont ils perceptibles? Le but de ces cartes d'interpolations étant d'obtenir les distributions régionales des taux de pollens d'arbres, nous viserons des cartes d'interpolations montrant le moins d'effets locaux possible (effets de lissage).

1.3 Imaginer un protocole permettant de calculer une grille de densité archéologique (ex. pour chaque pixel, attribution du nb de sites archéologiques situés dans un rayon de 10km)

- utilisez pour cela la carte de points archéologiques (*bdsitearcheoabcorrige_wgs84.shp)* (cf. § Boite à outils, cartes de densité)

1.4 Générer des points répartis de façon aléatoire et attribuer à ces points les informations des différents paramètres (altitudes, AP interp..)

- utiliser les outils d'intersection d'un SIG (cf. § Boite à outils)

Rq. Utilisez les départements 22, 29, 56, 35 et 44 comme surface pour établir le nuage de points. (100 pt par département).

Rq. En cas de message d'erreur pour cause de géométrie non valide, pour cela paramétrer Qgis pour passer l'erreur. Aller dans l'onglet Préférence → Option → Traitement → Général → modifier "Arrêter l'execution de l'algorithme quand une géométrie est invalide" et "Passer les entités avec des géométrie invalides".

1.5 Attribuer une information de distance à la mer(cf. § Boite à outils)

2. Analyses statistiques des données

2.1 Comparaison des paramètres deux à deux :

- la distance à la mer et le taux de pollens d'arbres,

- la distance à la mer et le nombre de sites archéologiques

- le taux de pollens d'arbres et le nombre de sites archéologiques

Rq. Voir l'utilisation de la fonction pairs() dans R. (exercice des ACP)

- Faire les graphiques et calculer les différents coefficients (corrélations, déterminations)

- Tester différentes droites et courbes de régressions

2.2 Faire une Analyse en Composantes Principales sur les 1000 objets(cf. § Boite à outils)

- Faire une ACP avec toutes les variables actives puis en mettant la longitude et la latitude en variables illustratives

2.3 Un mot sur les analyses de co-inertie

3. Autres exercices possibles :

- cartes d'interpolations d'autres périodes (cf. le fichier *bdculture+bdsite.shp) ex. champ NF_AP; champ AF_AP; Champ GR_AP;*

- Faire un test des quadrats sur la carte des points archéologiques

- Faire une AFC à partir des données botaniques (en P/A) identifiées dans les niveaux recoupant l'Age du Bronze (fichier : */Donnees/Donnees_Exercices/Exercice1_AB/table_pour_AFC_vegetation_AB.ods*)

Exercice 2 : Interpolation de valeurs d'affaiblissements (#débit internet) pour le département Haute Garonne (31).

Les points fournis sont issus de tests de débits (affaiblissements) d'un fournisseur internet. Les valeurs que nous allons chercher à interpoler sont des valeurs d'affiblissements qui sont proportionnelles à l'intensité du débit internet.

Nous disposons du fichier : nra_dept31_WGS84.shp (projection WGS84)

2.1 Afficher les points et reprojeter les points en Lambert 93

Reprojeter les données de façon à obtenir les coordonnées des points en Lambert93 (ou Lambert2) en mètres car les données d'origine sont en degrés Lat/Long. Retirer les points qui sont éloignés du département. Retenir peut-être qu'un sous-ensemble des 89000 points très restreint (commencer avec #100points).

2.2 Reprendre les protocoles des différentes interpolations

2.2.1 Faites un test des quadrats sur la distribution des points

- Faire des tests avec des disques de 5km, 10km, 50km
- Modifier la fenêtre

2.2.2 Interpolations sur l'ensemble des points

- Spline, IDW, (Krigeage?)

RQ. L'interpolation par krigeage semble poser pb..., il y a une erreur systématique au moment de l'utilisation de Predict() in gstat Warnings à l'ensemble des points : "Covariance matrix singular at location […]). Malgré l'utilisation de coordonnées en Lambert 93 ou Lambert2. Le problème est probablement lié au nb d'observations. Tester avec 50 à 100 points seulement sur une fenêtre géographique restreinte?

- Discuter les résultats (tenue de l'impact ou non des valeurs exceptionnelles, de la proximité des points, du lissage)

2.2.3 Interpolations après avoir retiré les points trop éloignés du département (31)

- Spline, IDW, (Krigeage)
- Discuter les résultats

2.3 Faire des suggestions d'analyses, les mettre en place?

-> comparaison au réseau routier, à l'altitude?..

Exercice 3 : Interpolation de valeurs d'ondes éléctromagnétiques obtenues dans les rues de Rennes métropôle

Il s'agit d'une couche de points. Chaque point correspond à une mesure de la puissance des ondes radios GSM et UMTS mais uniquement sur les fréquences utilisées par les opérateurs de téléphonie mobile (900 MHz, 1800 MHz et 2100 MHz).

Les mesures ont été réalisées par un appareil de mesure des ondes électromagnétiques monté sur un véhicule qui s'est déplacé dans les rues de Rennes durant 3 semaines en avril 2012.

Pour la carte des ondes, consultable sur http://www.georennes.fr/ondes-antennes/, seul l'attribut "field_av" a été exploité.

Les mesures sont exprimées en volts par mètre (V/m).

Système de référence original : RGF93 CC48 (IGNF : LAMBCC48 - EPSG : 3948)

2.1 Afficher les points et reprojeter les points en lambert 93

Reprojeter les données de façon à obtenir les coordonnées des points en Lambert93 (ou Lambert2) en mètres.

(cf. § boîte à outils)

2.2 Reprendre les protocoles des différentes interpolations

2.2.1 Faites un test des quadrats sur la distribution des points

- Faire des tests avec des disques de 5km, 10km, 50km

- Modifier la fenêtre

2.2.2 Interpolations sur l'ensemble des points

- Spline, IDW, Krigeage

- Discuter les résultats (la répartition des points suit les rues. Quel est l'impact sur les interpolations, la proximité des points?)

RQ. Les interpolations fonctionnent sur une sélection géographique de points restreinte (#100 points).

Exercice 4 : Recherche de dépendance entre des types de végétations et la pluviométrie

Faites un test de dépendance sur les deux séries de végétations : Celle où le chêne domine le noisetier et inversement en fonction des valeurs de précipitations.

Utilisez les données de couches de points de végétation (carte_pt_vegetation.shp) et de pluviométrie (Carte_Precipitation_Lamb2.shp) situées dans le repertoire Donnees/Donnees_Exercices/Exercice4_PRECIPITATION_VEGETATION.

Utilisez la projection Lambert2 (epsg : 27572)

1. Attribuer des valeurs de précipitation à l'ensemble des points.

2. Identifier les deux séries de valeurs à l'aide du champ "CHEN_SUP_N" (chene supérieur au noisetier =1; et l'inverse =0).

Utiliser onglet “Couche” → “Filtrer” pour sélectionner les deux séries de valeurs.

3. Récupérer les deux séries de valeurs numériques de précipitation et faire deux vecteurs de données à tester sur R.

3. Identifier la distribution des données (comparer la variance et la moyenne à l'aides des fonction mean() et var() sur R).

(comparer la variance et la moyenne de chaque série : les variances > moyennes, les données sont quantitatives mais les distributions ne sont pas de type "normale".)

4. Utiliser le test adequat pour montrer une différence signifactive ou non entre les deux séries de données. cf. Chapitre sur les tests et scripts sur les tests.

Chapitre 6 :Boîte à outils (pour QGis):

1. Génération de points aléatoires :

Vecteur->Outil de recherche -> Points aléatoires, nécessite une couche de polygones

Rq. Utilisez par exemple les départements 22, 29, 35 et 44 comme surface pour établir le nuage de points. (100 pt par département).

Rq. Lors de la génération des points il est probable que les coordonnées des points ne soient pas indiqués. Les créer. Pour cela il faut Editer la table attributaire (clic droit sur la couche, puis basculer en mode édition). Puis ouvrir la table attributaire, aller dans la console de Calculatrice de champs, créer deux nouveaux champs de type Réel, puis indiquer comme valeur d'expression, respectivement $x et $y.

2. Reprojection puis créer des champs avec les nouvelles coordonnées :

Reprojeter la couche en Lambert93 (ou lambert2) -> créer un nouveau shp (Reproject layer) puis créer 2 nouveaux champs X et Y à l'aide de l'outil d'édition sur la table attributaire et calculer les valeurs avec les coordonnées courantes $x et $y.

Voir aussi outil de géométrie -> Exporter / Ajouter des colonnes. Jouer sur la projection du projet et celle de la couche pour obtenir 2 nouvelles colonnes avec les coordonnées reprojetées.

3. Créér des disques :

Créer à partir de vecteurs/points des tampons ou buffers

4. Comptage du nb de points dans chaque disque :

Outil d'analyse : Points dans un polygone.
On génère ainsi une couche temporaire avec un champ de comptage (NUMPOINTS). Il est possible d'utiliser la vue “statistique” (Onglet Vue-> Résumé statistique) pour afficher toutes les caractéristiques statistiques de ce champ. On obtient ainsi directement la moyenne et l'écart-type (Variance) de ce champ.

5. Jointures

Pour faire une jointure attributaire double clic droit sur la couche, onglet jointures. Il y a moyen d'éditer la jointure en bas de fenêtre. Bouton “+”, “-”, et crayon pour éditer.

6. Pour attribuer de l'information d'un raster à des points :

- si l'on part d'un raster (grille) il faut transformer le raster en polygones (Raster->Conversion->Polygoniser). Veiller à ce que l'amplitude des valeurs de la grille soit suffisante (de l'ordre de 100) pour obtenir une couche vecteur avec suffisamment d'unités.

- extraire l'information en faisant une intersection entre la couche de points et la couche de polygones issue du raster.

Outils de géotraitement -> Intersection

Rq. Pour l'intersection attention à bien être dans les mêmes systèmes de projection au niveau de chacune des couches comparées. Pas seulement au niveau de l'affichage mais au niveau de la projection native de la couche! Au besoin il faut reprojeter la couche avec le script "Reproject raster layer" (dans Traitement -> Boîte à outils). Les valeurs des polygones seront attribuées à la couche de points.

Rq. En cas de message d'erreur pour cause de géométrie non valide, pour cela paramétrer Qgis pour passer l'erreur. Aller dans l'onglet Préférence → Option → Traitement → Général → modifier "Arrêter l'execution de l'algorithme quand une géométrie est invalide" et "Passer les entités avec des géométrie invalides".

7. Pour attribuer une information de distance :

Rechercher des distances entre des points et une polyligne. On va pour cela transformer la polyligne en points sinon le calcul de distance est effectué avec le barycentre de la polyligne :

- Outil de géometrie->extraire les sommets (ex. Sur la ligne de trait de côte avec le fichier *ligCLC.shp*)

- Utilisation ensuite d'un scripts QGis : (existance de surcouche à Grass ou R..)

-> utilisation de "Distance to nearest hub" (ou Distance au plus proche centre - points) (script R) à chercher sur R si on souhaitait aller plus loin.

- source = la couche de points

- destination = la couche de points issue de la polyligne

On recupère une couche "sortie garphique" avec les paramètres de distance "HubDist".

Rq. Sur Postgis (base de données) : existe une fonction st_distance()

8. Générer une carte de densité

- Le plus simple (sans carte de densité) : création pour chaque point d'observation d'un tampon de 10 km à partir duquel on compte le nombre le nb de points recherchés (archéologiques) avec : outil analyse -> point dans un polygone

ex. pour chaque point généré de façon aléatoire on calcule du nb de points compris dans un périmètre de 10km

- Autre suggestion : calcul pour chaque pixel d'une grille, du nombre de points recherchés dans un rayon défini. C'est le principe de l'estimation par noyau. Le protocole est accessible grâce à l'outil de traitement “heatmap” ou “carte de chaleur”. Vous pouvez le retrouver en utilisant la recherche dans Qgis. Indiquez la carte de points vecteur en entrée (attention à bien définir l'emprise souhaitée). Indiquez le rayon de la recherche (ex. 10000 mètres autour de chaque pixel). Paramétrez la taille du pixel du raster en sortie (ex. 3000 mètres, bien indiquer la même unité que le rayon de recherche). Indiquez un “Kernel Shape” uniforme (sans transformation des valeurs comptées).

9. Analyse en Composantes Principales :

En R, l'analyse en composantes principales est disponible via la fonction princomp() du package stats :

Objectifs :

- Réaliser une ACP sur un fichier de données.

- Afficher les valeurs propres. Construire le graphiques éboulis des valeurs propres.

- Construire le cercle de corrélations.

- Projeter les observations dans le premier plan factoriel.

9.1 Chargement des données

```
dataACP<-
read.csv(file="/Donnees/Donnees_Chapitre_4/Exercice1_AB/t
ablepourACP.csv",header=TRUE,sep=";",dec=",");
```

#Quelques vérifications :

```
print(dataACP)
```

#statistiques descriptives
```
summary(dataACP)
```

#nuages de points : peut être intéressant afin de percevoir déjà des corrélations
```
pairs(dataACP)
```

#partition des données (var. actives et illustratives). La première colonne correspond aux identifiants elle n'est donc pas prise en compte
```
dataACP.actifs <- dataACP[,2:8]
```

il n'y a pas de valeurs illustratives on aurait pu mettre les valeurs de longitude et latitude. Rq. Refaire l'ACP en mettant les longitudes et latitudes en val illustratives?
```
#dataACP.actifs <- dataACP[,4:8]
#dataACP.illus <- dataACP[,2:3]
```

#nombre d'observations
```
n <- nrow(dataACP.actifs)
print(n)
```

#Utilisation de **princomp()**

centrage et réduction des données --> cor = T (cette étape consiste en une conversion de chaque paramètre vers un standard commun ou le fait de rendre quelque chose conforme à un standard. Cette transformation rendra toutes les valeurs (indifféremment de leurs distributions et unités de mesures originales) en unités compatibles.

- Réduire une variable consiste à diviser toutes ses valeurs par son écart type.
- Centrer une variable consiste à soustraire son espérance à chacune de ses valeurs initiales, soit retrancher à chaque donnée la moyenne (c'est ce qui s'appelle un centrage). Elle constitue simplement en un changement d'origine, qui place la moyenne de la distribution au point 0 de l'axe des abscisses.
- L'espérance d'une variable correspond à une moyenne pondérée des valeurs que peut prendre cette variable.

En bref, avec cette étape on créé des variables (ou composantes) qui vont servir de standard. Cela va correspondre aux axes des analyses mulivariées sur lesquels les valeurs de chaque point sont projetées.

#calcul des coordonnées factorielles --> scores = T

```
acp.dataACP <- princomp(dataACP.actifs, cor = T, scores =T)
```

#print : on obtient les écarts types associés aux axes. Typiquement l'axe 1 est celui qui présente le plus de variabilité (variance).

```
print(acp.dataACP)
```

#summary

```
print(summary(acp.dataACP))
```

exemple :

```
> print(acp.dataACP)
Call:
princomp(x = dataACP.actifs, cor = T, scores = T)
Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
1.7017084 1.2207731 1.0061553 0.8338799 0.7164325 0.4894175 0.3916532
 7  variables and  5315 observations.
> print(summary(acp.dataACP ))
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5    Comp.6     Comp.7
Standard deviation     1.7017084 1.2207731 1.0061553 0.83387990 0.71643247 0.4894175 0.39165321
Proportion of Variance 0.4136874 0.2128981 0.1446212 0.09933653 0.07332507 0.0342185 0.02191318
Cumulative Proportion  0.4136874 0.6265855 0.7712067 0.87054326 0.94386832 0.9780868 1.00000000
```

On obtient les écart-types associés aux axes. Le carré des ecart-types (ou standard deviations) correspond aux variances = valeurs propres. Nous avons également le pourcentage cumulé. Nous obtenons pres de 77% de la variance cumulée avec les 3 premières composantes.

#quelles sont les propriétés associées à l'objet ? Avec ATTRIBUTES, nous avons la liste des informations exploitées par la suite.

```
print(attributes(acp.dataACP))
```

```
$names
[1] "sdev"     "loadings" "center"   "scale"    "n.obs"    "scores"   "call"
$class
[1] "princomp"
```

9.2 Valeurs propres

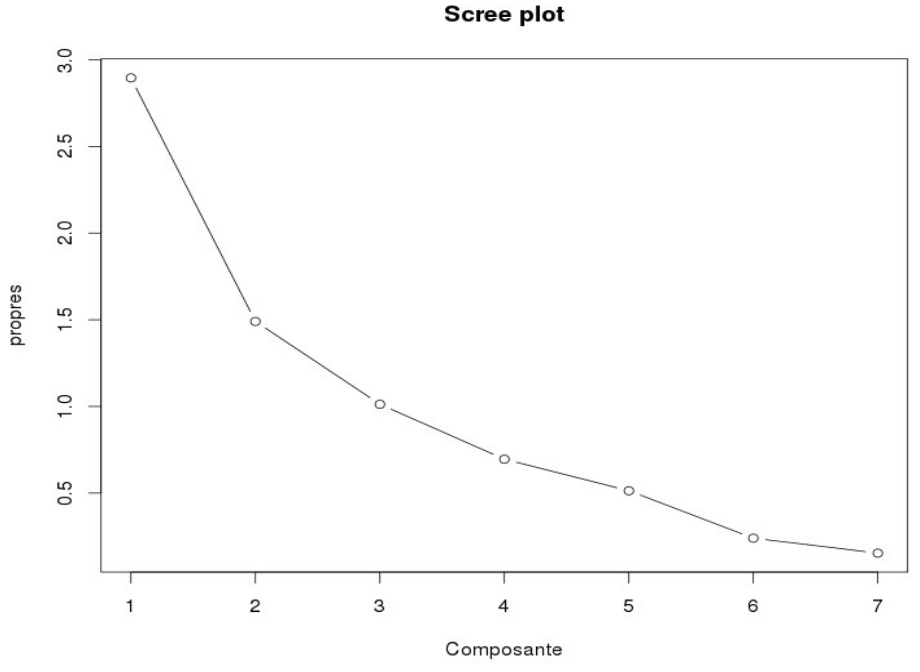
#obtenir les variances associées aux axes c.-à-d. les valeurs propres (on les obtient en mettant les écart-types au carré)

```
val.propres <- acp.dataACP$sdev^2
print(val.propres)
```

```
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
2.8958115 1.4902870 1.0123486 0.6953557 0.5132755 0.2395295 0.1533922
```

#Graphique des éboulis des valeurs propres

```
plot(1:7,val.propres,type="b",ylab="Valeurs
```




```
propres",xlab="Composante",main="Scree plot")
```

Exemple pour la composante 1 : 1.7017084² = 2,895811

Les 2 premiers axes représentent près de 63% de l'information disponible. Le troisième axe contient aussi une part non négligeable de l'information (environ 14%).

9.3 Cercle de corrélations : variables actives

Recherche de corrélations variables-facteurs
c1 = composante1; c2 = composante2
On calcule les coordonnées des vecteurs sur les axes C1 et C2

```
c1 <- acp.dataACP$loadings[,1]*acp.dataACP$sdev[1]
c2 <- acp.dataACP$loadings[,2]*acp.dataACP$sdev[2]
```

#affichage : on va calculer les coordonnées des variables sur les composantes 1 et 2 qui vont correspondre aux axes

```
correlation <- cbind(c1,c2)
print(correlation,digits=2)
```

#carrés de la corrélation (cosinus2) :

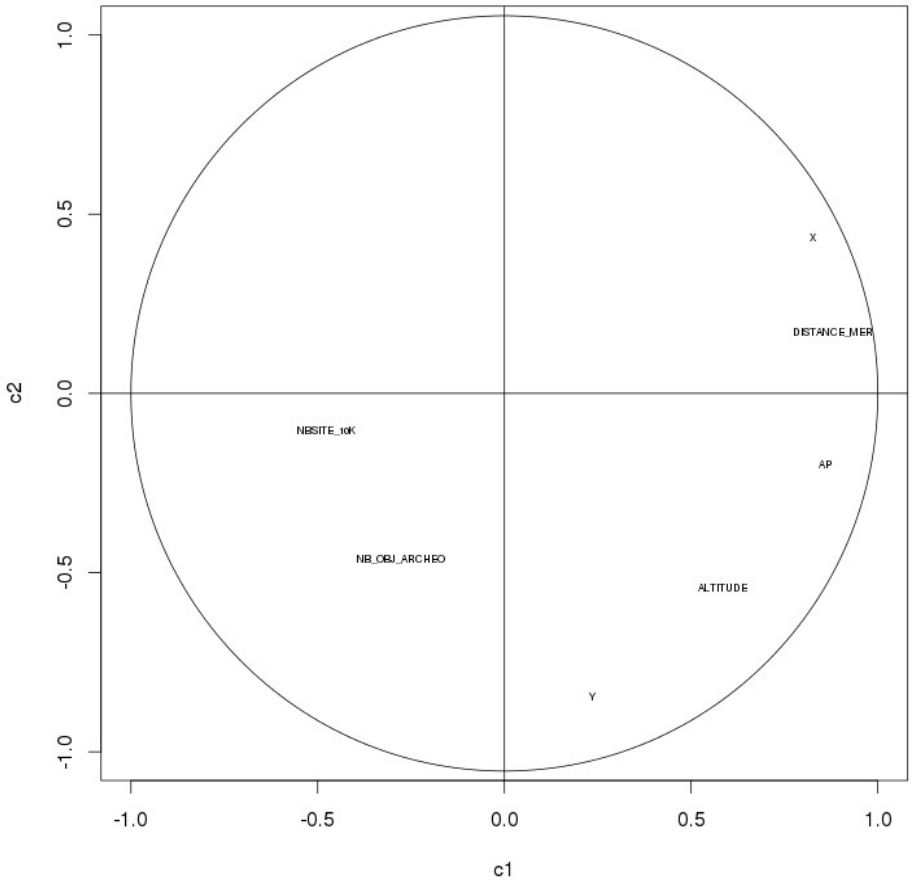
```
print(correlation^2,digits=2)
```

#cumul carrés de la corrélation

```
print(t(apply(correlation^2,1,cumsum)),digits=2)
```

#*** cercle des corrélations - variables actives ***

```
plot(c1,c2,xlim=c(-1,+1),ylim=c(-1,+1),type="n")
abline(h=0,v=0)
text(c1,c2,labels=colnames(dataACP.actifs),cex=0.5)
symbols(0,0,circles=1,inches=F,add=T)
```

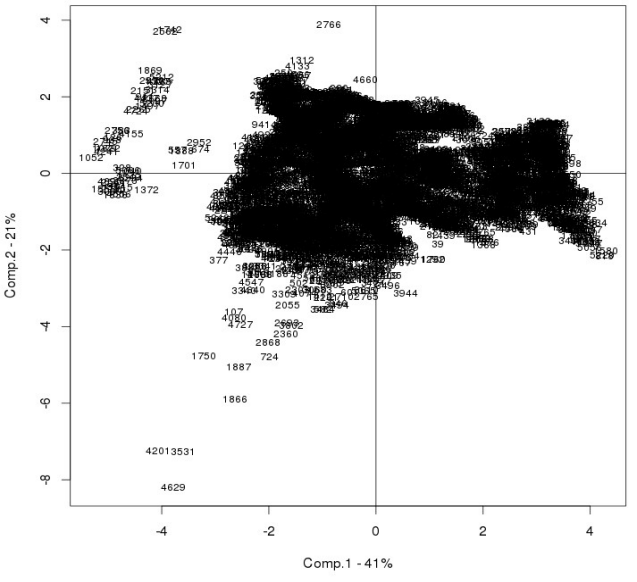


Cercle des corrélations : l'axe c1 est expliqué principalement par la distance à la mer (DISTANCE_MER) (=continentalité), le taux de pollens d'arbres fossiles (AP) et le nombre de site archéologiques (NBSITE_10km). L'axe2 semble davantage être expliqué par la latitude (Y).

9.4 Carte des individus (objets) dans le premier plan factoriel (63% de l'inertie)

#l'option "scores" demandé dans princomp est très important ici

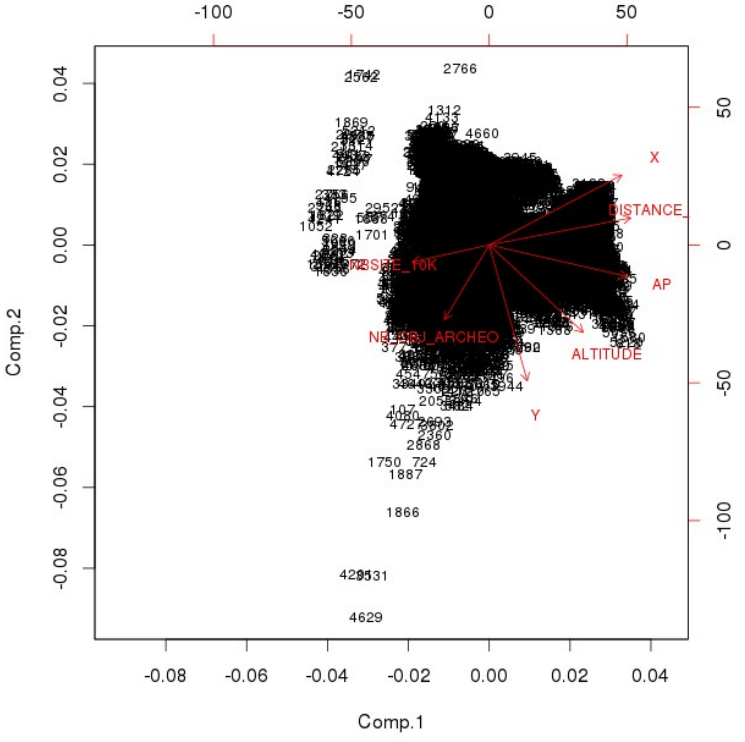
```
plot(acp.dataACP$scores[,1],acp.dataACP$scores[,2],type="n"
,xlab="Comp.1 - 41%",ylab="Comp.2 - 21%")

abline(h=0,v=0)

text(acp.dataACP$scores[,1],acp.dataACP$scores[,2],labels=r
ownames(dataACP.actifs),cex=0.75)
```



Carte des individus ou objets multiparamétrés dans le premier plan factoriel. Ce plan peut être particulièrement intéressant pour localiser des valeurs particulières.

9.4 Représentation simultanée : individus x variables

```
biplot(acp.dataACP, cex=0.75)
```



Dans ce graphique il faut avant tout s'attacher aux directions des vecteurs par rapport aux individus.