Département de Physique
Master spécialisé Master Intelligence Artificielle et Réalité Virtuelle

Faculté des Sciences
كلية العلوم

Rapport réalisé par:
Zeghli Fatima Zahra
Ait Oumar Ouafa
Bourass Wiame

Supervisé par:
Pr. Kaicer mohammed

Titre:

« Les normes et les distances au sens mathématiques »

Année Universitaire : 2022/2023

I- La norme

1- Définition de la norme au sens mathématique:

Une norme au sens mathématique est une fonction qui permet de mesurer la taille ou la distance entre des éléments d'un espace vectoriel. Elle est généralement notée comme ||x||, où x est un élément de l'espace vectoriel considéré.

2- Les propriétés d'une norme:

Pour être considérée comme une norme, cette fonction doit respecter certaines propriétés :

- ➔ Positivité : la norme d'un élément doit être positive, c'est-à-dire que ||x|| >= 0 pour tout x dans l'espace vectoriel, et ||x|| = 0 si et seulement si x = 0 (c'est-à-dire que la norme de l'élément nul est nulle)
- ➔ Homogénéité : la norme doit être homogène, c'est-à-dire que ||ax|| = |a|.||x|| pour tout réel a et tout x dans l'espace vectoriel (c'est-à-dire que la norme d'un élément est proportionnelle à la norme de son coefficient multiplicateur)
- ➔ Sub-additivité : la norme doit être sub-additive, c'est-à-dire que ||x ▪ y|| <= ||x|| ▪ ||y|| pour tout x et y dans l'espace vectoriel (c'est-à-dire que la norme de la somme de deux éléments est inférieure ou égale à la somme des normes de ces éléments)
- ➔ Triangulaire : la norme doit respecter la propriété triangulaire, c'est-à-dire que ||x ▪ y|| <= ||x|| ▪ ||y|| pour tout x et y dans l'espace vectoriel.
- ➔ Continuité : la norme doit être continue, c'est-à-dire qu'elle ne change pas brusquement lorsque l'on modifie légèrement les éléments de l'espace vectoriel.
- ➔ Invariance par rotation : la norme doit être invariante par rotation, c'est-à-dire qu'elle ne change pas lorsque l'on effectue une rotation sur l'espace vectoriel.
- ➔ Invariance par translation : la norme doit être invariante par translation, c'est-à-dire qu'elle ne change pas lorsque l'on effectue une translation sur l'espace vectoriel.
- ➔ Invariance par homothétie : la norme doit être invariante par homothétie, c'est-à-dire qu'elle ne change pas lorsque l'on effectue une homothétie sur l'espace vectoriel.

**Une homothétie est une transformation géométrique par agrandissement ou réduction ; autrement dit, une reproduction avec changement d'échelle. Elle se caractérise par son centre, point invariant, et un rapport qui est un nombre réel. Par l'homothétie de centre O et de rapport k, le point M est transformé en un point N tel que:*

$$\overrightarrow{ON} = k\overrightarrow{OM}.$$

3- Les types des normes

Il existe plusieurs types de normes, dont les plus courantes sont :

- La **norme Euclidienne** : c'est la norme classique utilisée en géométrie euclidienne, qui est définie comme la racine carrée de la somme des carrés des composantes d'un vecteur. Pour un vecteur $v = (v_1, v_2, ..., v_n)$ dans R^n, la norme euclidienne est donnée par:

$$N(v) = \sqrt{(v_1^2 + v_2^2 + ... + v_n^2)}$$

- La **norme de Manhattan**: également appelée norme Ll, est une autre norme couramment utilisée dans l'espace vectoriel des vecteurs à coordonnées réelles ou complexes. Elle est définie comme la somme absolue des coordonnées d'un vecteur. Pour un vecteur $v = (v_1, v_2, ..., v_n)$dans R^n, la norme de Manhattan est donnée par :

$$N(v) = |v_1| + |v_2| + ... + |v_n|$$

- La **norme infinie**: nommée aussi L-infini est une norme couramment utilisée dans l'espace vectoriel des vecteurs à coordonnées réelles ou complexes. Elle est définie comme la valeur absolue la plus grande des coordonnées d'un vecteur. Pour un vecteur $v = (v_1, v_2, ..., v_n)$dans R^n, la norme infinie est donnée par :

$$N(v) = max(|v_1|, |v_2|, ..., |v_n|)$$

II- Distance

1- Définition d'une distance mathématique:

En mathématiques, une distance est une fonction qui mesure la "distance" entre deux éléments d'un ensemble. Plus précisément, une distance sur un ensemble E est une fonction $d : ExER$ qui satisfait les propriétés suivantes :

- $d(x, y) \geq 0$ pour tout $\forall$ x, y dans E.
- $d(x, y) = 0$ si et seulement si x = y pour tous x, y dans E.
- $d(x, y) = d(y, x)$ pour tous x, y dans E (symétrie).
- $d(x, z) \leq d(x, y) * d(y, z)$ pour tous x, y, z dans E (propriété de triangularité).

La dernière propriété est souvent appelée "inégalité triangulaire" ou "propriété de la distance".

2- Définition d'une distance statistique:

La distance statistique est un concept utilisé en statistiques et en analyse de données pour mesurer la similarité ou la dissimilarité entre deux éléments. Il existe plusieurs types de distances statistiques, chacun ayant des propriétés différentes et étant adapté à des types de données spécifiques.

La distance mathématique et la distance statistique sont deux concepts différents qui sont utilisés dans des domaines différents.

3- Différence entre distance mathématique et distance statistique:

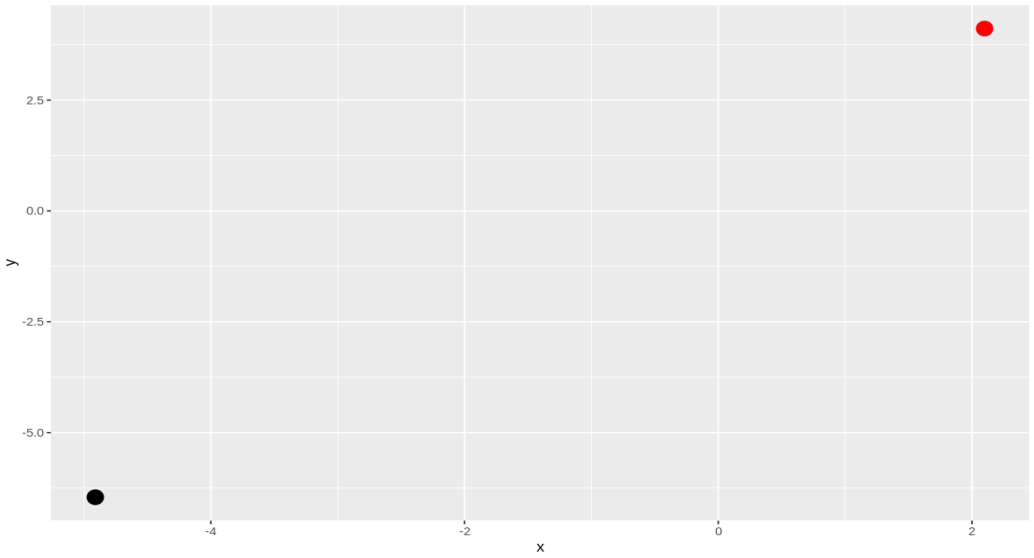
La distance mathématique est généralement utilisée pour mesurer la distance physique entre deux points dans l'espace. La distance statistique, d'autre part, est utilisée pour mesurer la similarité ou la dissimilarité entre des éléments dans un jeu de données. Il est souvent utilisé dans des applications telles que la classification, le regroupement et l'analyse en composantes principales. Mais les deux distances peuvent être utilisées ensemble ou séparément selon les besoins de l'application.

4- Introduction aux quelques distances métriques:

Avant de décrire chaque concept de distance, en premier on pose deux points $point1, point2$ aléatoires sur notre plan (x, y) avec le code R suivant:

```
# Générer deux points aléatoires
x1 <- runif(1, -10, 10)
y1 <- runif(1, -10, 10)
x2 <- runif(1, -10, 10)
y2 <- runif(1, -10, 10)

point1 <- data.frame(x1, y1)
point2 <- data.frame(x2, y2)
library(ggplot2)
ggplot() + geom_point(data = point1, aes(x1, y1), size = 5 ,color = 'red') +
geom_point(data = point2, aes(x2, y2), size = 5)
```

On aura l'exécution suivante:



a- distance : Manhattan Distance et Minkowski Distance

i- Manhattan Distance

Manhattan distance (also known as "taxi cab" or "Ll" distance) is a measure of the distance between two points in a multi-dimensional space. It is calculated as the sum of the absolute differences of the coordinates of the two points.
Généralement la formule dans un espace multi-dimensionnels entre deux points $A(a_1, a_2, a_3, ..., a_n)$ et $B(b_1, b_2, b_3, ..., b_n)$ est:

$$D_m(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Pour notre exemple l'espace est deux dimensionnels on a (x_1, y_1) and (x_2, y_2) d'où:

$$D_m(point1, point2) = |x_1 - x_2| + |y_1 - y_2|$$

ii- Minkowski Distance

La distance de Minkowski est une mesure de distance générique qui peut être utilisée pour définir la distance entre deux points dans un espace vectoriel de n dimensions. Elle est définie comme la racine n-ième de la somme des différences entre les coordonnées des deux points, élevées à la puissance p.
La formule générale est :

$$D_m(x, y) = (\sum_{i=1}^{n} |xi - yi|^p)^{(1/p)}$$

Lorsque p = 2, cette distance est égale à la distance Euclidienne classique, et lorsque p = 1, elle est égale à la distance de Manhattan.
Pour notre exemple l'espace est deux dimensionnels on a (x_1, y_1) and (x_2, y_2) d'où:

$$D_m(point1, point2) = [(|x_1 - y_1|^p) + (|x_2 - y_2|^p)]^{(1/p)}$$

iii- Méthode de Manhattan en r

```
x_diff <- abs(point1$x - point2$x)
y_diff <- abs(point1$y - point2$y)
manhattan_distance <- sum(x_diff, y_diff)
paste("Manhattan Distance between point1 and point2 is equal to
",manhattan_distance)
```

'Manhattan Distance between point1 and point2 is equal two points = 19.29'

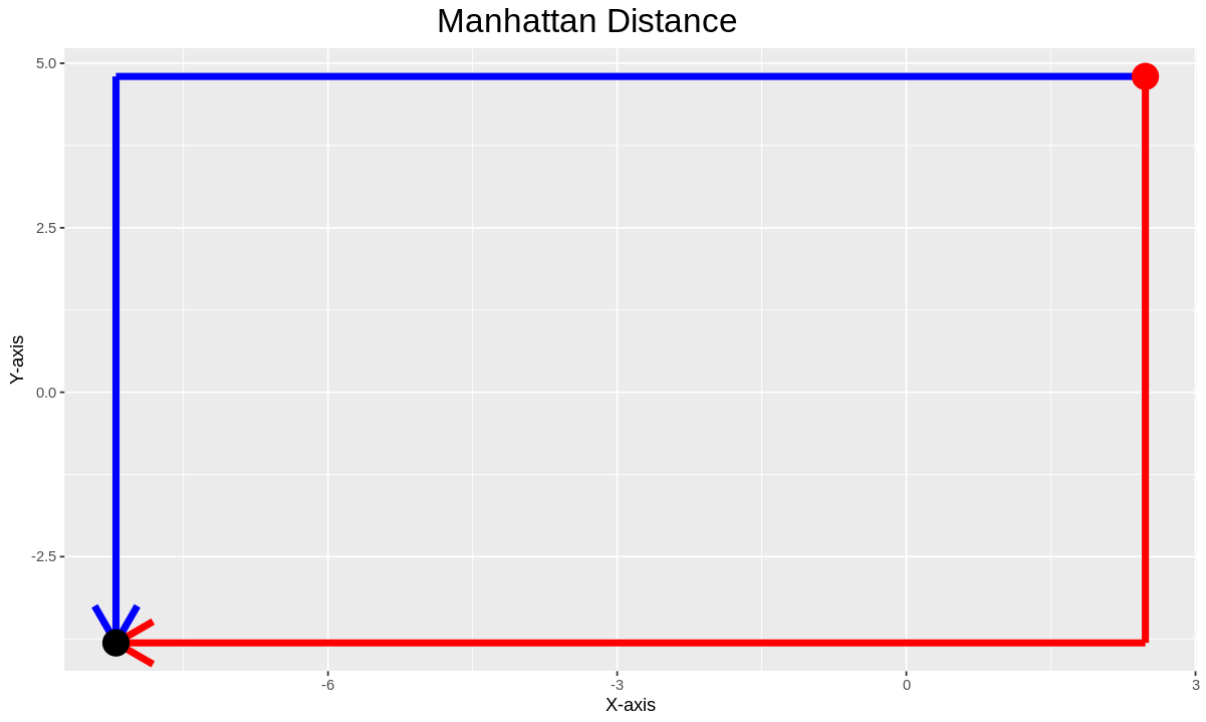
iv- Méthode de Minkowski en r

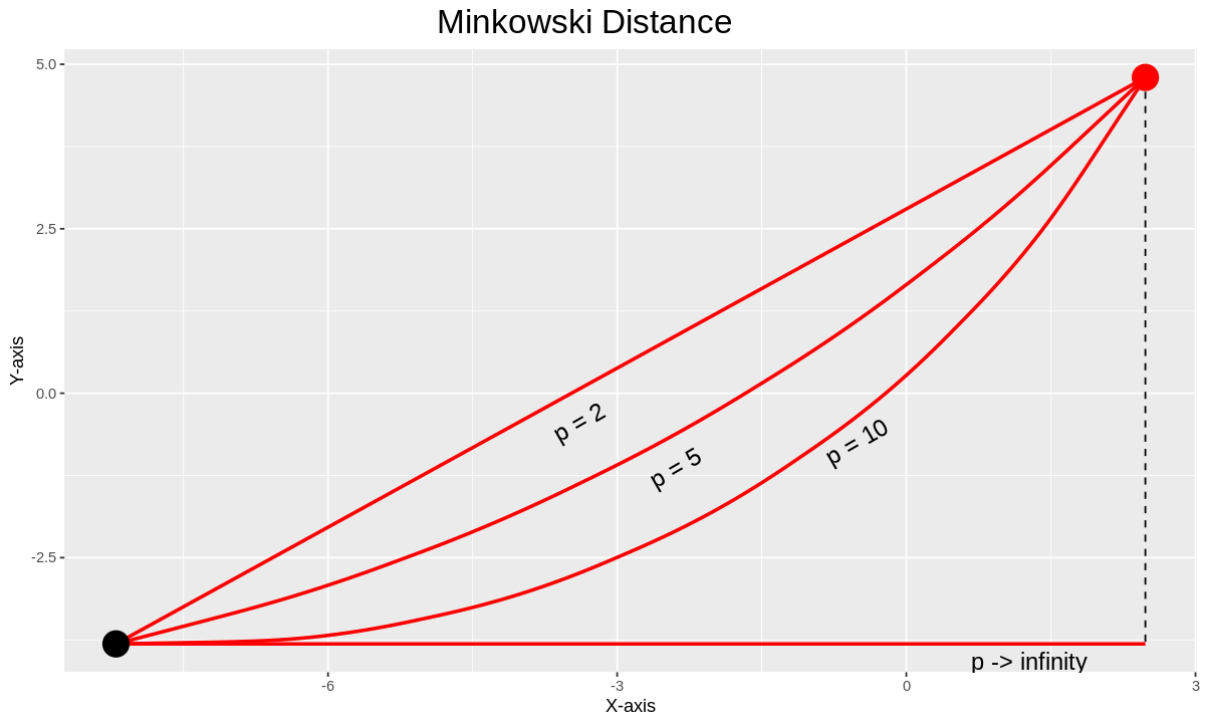
En changeant la valeur de p on constate qu'avec p=1 on se trouve avec la valeur initial de Manhattan et comme p augmente le résultat tend vers l'infini d'où la distance est jamais atteinte.

```
pvalues <- c(1,2,100,200,1000)
for(i in 1:(length(pvalues))) {
m = dist(rbind(c(x1, y1), c(x2, y2)), method = "minkowski", p =  pvalues[i])
cat(sprintf("Minkowski Distance avec p = %d est %.2f\n",  pvalues[i],m))}
```

```
Minkowski Distance avec p = 1 est 19.29
Minkowski Distance avec p = 2 est 13.72
Minkowski Distance avec p = 5 est 11.32
Minkowski Distance avec p = 10 est 10.80
Minkowski Distance avec p = 100 est 10.68
Minkowski Distance avec p = 200 est 10.68
Minkowski Distance avec p = 1000 est Inf
```

v- Visualisation des deux méthodes en r





b- distance : Mahalanobis Distance et Jaccard distance

i- Mahalanobis Distance

La distance de Mahalanobis en statique est un outil utilisé pour mesurer la distance entre un point de données et la moyenne de la distribution de données dans un espace multidimensionnel. Elle prend en compte la covariance entre les différentes variables et permet de détecter les outliers dans les données. Cette distance est calculée en utilisant la matrice de covariance inverse de l'ensemble des données. La formule générale pour calculer la distance Mahalanobis entre un point x et la distribution de référence (avec moyenne vectorielle μ et matrice de covariance Σ) est donnée par :

$$d(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

ii- Jaccard Distance

La distance de Jaccard est utilisée pour mesurer la similarité entre deux ensembles. Elle est définie comme étant le rapport entre la taille de l'intersection des deux ensembles et la taille de l'union des deux ensembles. Formellement, la distance de Jaccard est définie comme étant :

$$J(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

où A et B sont les deux ensembles à comparer, |A| est la taille de l'ensemble A, et |A ∩ B| est la taille de l'intersection de A et B.

iii- Méthode de Mahalanobis en r

```
# Generate random data
set.seed(123)
data <- data.frame(x1 = rnorm(100), x2 = rnorm(100), x3 = rnorm(100))

# Calculate covariance matrix
cov_matrix <- solve(cov(data))

# Calculate Mahalanobis distance for every point
mahal_dist <- mahalanobis(data, center = colMeans(data), cov = cov_matrix)

# Print distances
print(mahal_dist)
```

Quelques valeur du tableau:

-0.491031166	-0.21538051	0.84501300
-2.309168876	0.06529303	0.96252797
1.005738524	-0.03406725	0.68430943
-0.709200763	2.12845190	-1.39527435
-0.688008616	-0.74133610	0.84964305
1.025571370	-1.09599627	-0.44655722
-0.284773007	0.03778840	0.17480270
-1.220717712	0.31048075	0.07455118
0.181303480	0.43652348	0.42816676
-0.138891362	-0.45836533	0.02467498
0.005764186	-1.06332613	-1.66747510
0.385280401	1.26318518	0.73649596
-0.370660032	-0.34965039	0.38602657
0.644376549	-0.86551286	-0.26565163
-0.220486562	-0.23627957	0.11814451

Après avoir calculer la distance Mahalanobis, on peut savoir les outliers ou les cas particuliers de cette dataset.

Avec ce segment de code on conclut les outliers on créons un vecteur logique qui est 'TRUE' pour les lignes où la valeur de 'distance' est supérieure à l'échantillon de 80% et 'FALSE' sinon.

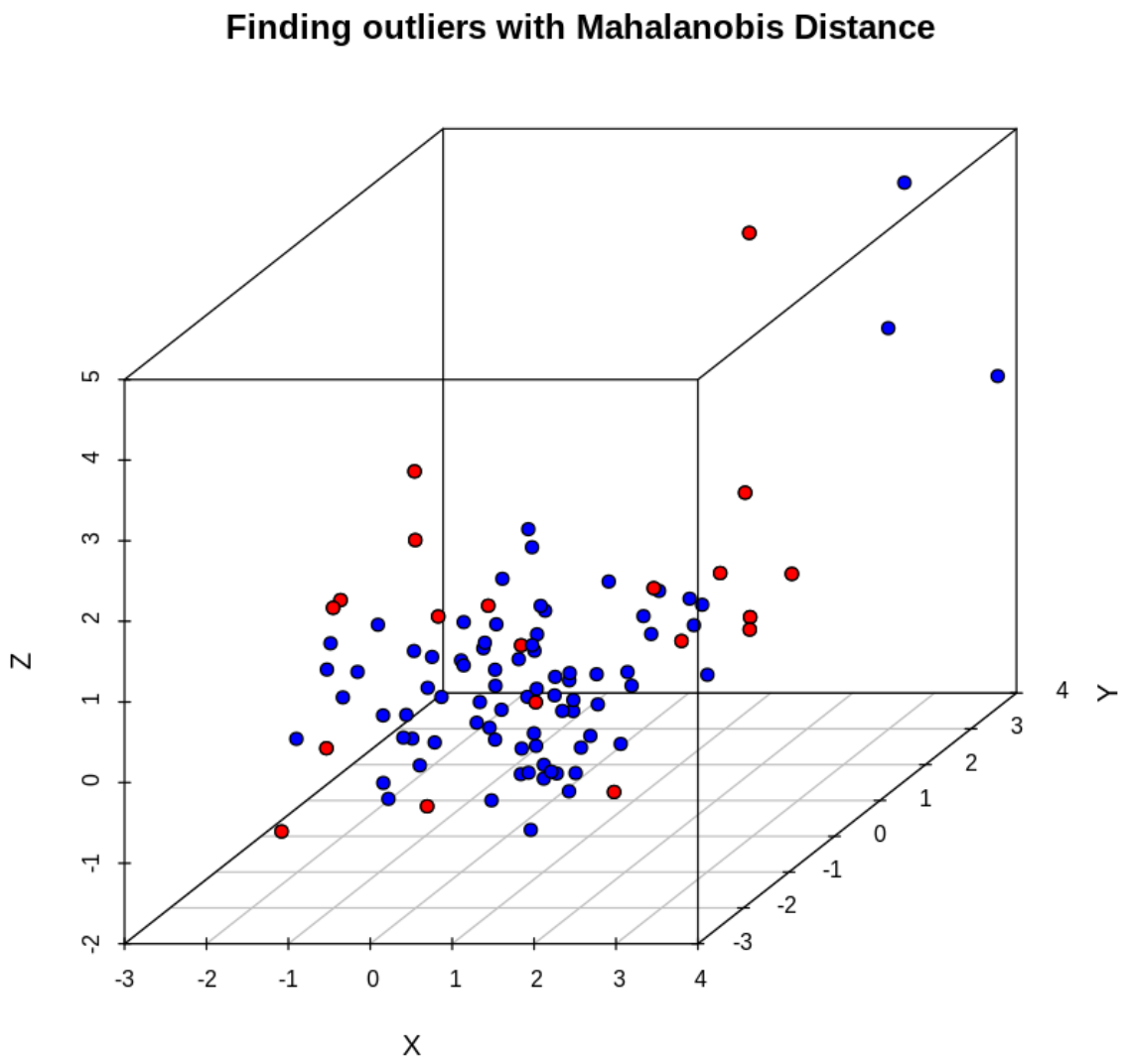
```
outliers <- data[data$distance > quantile(data$distance, 0.80),]
outliers
```

A data.frame: 20 × 5

	x	y	z	distance
	<dbl>	<dbl>	<dbl>	<dbl>
1	1.18529291	3.19654978	4.06528489	4.832682
5	2.36871505	0.97389042	1.83023207	5.572829
6	0.90759177	2.20532606	-0.55708038	4.897252
13	1.47743009	-2.02362702	0.56223381	6.207813
21	-1.36184763	2.04350430	-0.04733266	6.481440
30	-1.15841660	0.05810584	2.50176510	11.185699
34	0.05720034	-1.61012302	1.44276824	5.715831
40	1.83802787	1.38059133	0.65189100	4.796815
48	2.15652982	-1.52463454	-0.77306692	6.209442
52	-0.67253669	-2.75360875	0.31578806	6.254377
61	2.16566299	1.44870575	0.07504484	4.864244
67	-0.99757081	0.04551142	-1.64807617	6.061096
69	2.59949171	0.65970328	0.27211081	5.624840
72	-1.56806915	-2.12723268	-0.99443498	7.132698
76	2.89485439	1.05367298	0.78704034	8.680203
79	1.83265772	-0.06970099	1.10938000	5.257992
88	-1.47545512	2.97158503	-0.95133863	13.256937
91	-2.17399643	0.09627125	0.79157269	5.298201
93	-2.30479535	0.49523695	0.71031471	6.108905
96	-1.42847459	0.55982377	1.42649174	5.490284

iv- Visualisation des outliers trouver par la méthode Mahalanobis:

```
data$color <- 'blue'

outliers$color <- 'red'

m <- rbind(outliers,data)

W <- m$x
H <- m$y
D <- m$z
C <- m$color

scatterplot3d(x = W, y = H, z = D,  pch = 21, bg = C,
              xlab = "X", ylab = "Y", zlab = "Z" ,
              main = "Finding outliers with Mahalanobis Distance")
```



v- Méthode de Jaccard en r

La définition de la fonction Jaccard en r sert a trouvé la similarité entre 2 ensembles utilisée pour les cas suivants:

- **Exploration de texte** : recherche de la similitude entre deux documents texte en fonction du nombre de termes utilisés dans les deux documents.
- **E-Commerce** : trouvez des clients similaires via leur historique d'achat à partir d'une base de données de vente de milliers de clients et de millions d'articles.
- **Systèmes de recommandation** : recherche de clients similaires en fonction des notes et des avis, par exemple, algorithmes de recommandation de films, recommandation de produit, recommandation de régime alimentaire, recommandations de mariage, etc.

Pour raison de simplification on choisit 2 ensembles A et B assez clair:

```
a  <- c(0, 1, 2, 5, 6)
b <- c(0, 2, 3, 4, 5, 7, 9)
```

L'introduction de notre fonction jaccard:

```
jaccard <- function(a, b) {
    intersection = length(intersect(a, b))
    union = length(a) + length(b) - intersection
    return (intersection/union)
}
jaccard(a, b)
```

```
0.333333333333333
```

c- Distance: Wasserstein or Kantorovich–Rubinstein Distance

i- Wasserstein Distance

La distance de Wasserstein, également connue sous le nom de distance de Monge-Kantorovich, est une distance utilisée pour mesurer la similitude entre deux distributions de probabilité. Elle est définie comme étant le coût minimal pour convertir une distribution en une autre, où le coût est mesuré en utilisant une fonction de coût donnée.

Mathématiquement, si X et Y sont des espaces de probabilité et $c(x, y)$ est une fonction de coût, la distance de Wasserstein entre les distributions P et Q sur X et Y est donnée par :

$$W(P, Q) = inf \int c(x, y) d\gamma(x, y)$$

où γ est un transport de probabilité de P vers Q, et inf signifie infimum (c'est-à-dire le plus petit élément de l'ensemble).

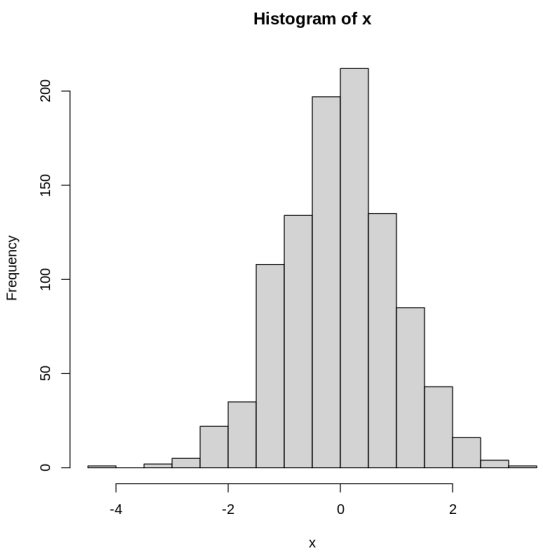
En pratique, il est souvent nécessaire de travailler avec des distributions discrètes plutôt que continues. Dans ce cas, la distance de Wasserstein peut être calculée en utilisant la méthode de transport optimal, qui consiste à trouver la matrice de transport qui minimise le coût total tout en respectant les contraintes de transport.

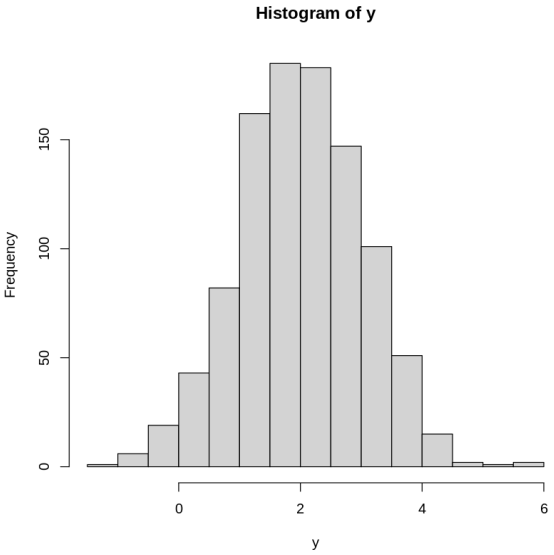
ii- Méthode de Wasserstein in r:

On génère deux ensembles d'échantillons de données normalement distribués, et crée des histogrammes des données, puis utilise la fonction wasserstein() de la bibliothèque waddR pour calculer la distance de Wasserstein entre les deux distributions. L'argument p dans la fonction wasserstein_metric() peut être ajusté pour modifier la puissance de la métrique de distance (la valeur par défaut est 2, qui est la distance standard de Wasserstein). La distance calculée est ensuite imprimée sur la console.

La fonction *wasserstein_metric*, offre une implémentation plus rapide en $C++$ de la fonction wasserstein1d du package R transport, qui est capable de calculer les distances p-Wasserstein générales. Pour p=2, on obtient la distance 2-Wasserstein W.

```
library(waddR)

# Generate sample data for two probability distributions
x <- rnorm(1000, mean = 0, sd = 1)
y <- rnorm(1000, mean = 2, sd = 1)

# Create histograms of the data
histx <- hist(x)
histy <- hist(y)

# Calculate the Wasserstein distance between the two distributions
wass_dist <- wasserstein_metric(histx$density, histy$density, p = 2)

# Print the calculated distance
print(wass_dist)
```






```
#> [1] 2.044457
```