Projet n°6: Réduction de dimension non-linéaire

Mamadou Lamine DIAMBAN

Avril 2019

Table des matières

Introduction

La réduction de dimension est la technique de représentation de données multidimensionnelles (des données à plusieurs caractéristiques ayant une corrélation entre elles) en 2 ou 3 dimensions. Généralement, l'Analyse en Composantes Principales (ACP) est l'algorithme de réduction de dimension linéaire le plus utilisé, lorque les variables sont quantitatives, mais est mal adaptée quand les données à analyser vivent dans un sous-espace présentant une géométrie non linéaire. C'est dans ce contexte que nous allons présenter une méthode de réduction de dimension non linéaire: t-sne (tStochastic Neighborhood Embedding) qui est maintenant une technique assez généralisée dans le domaine de l'apprentissage automatique. Contrairement à l'ACP qui tend à préserver les distances entre points distants, cette méthode vise à préserver les distances faibles dans la projection.
Ainsi nous travaillons avec deux jeux de données:
- `Swiss-Roll`: qui est constitué d'une matrice X en 3 dimensions et d'un vecteur y contenant la position des points dans le rouleau, et donc la couleur à leur associer.
- `Digits`: qui comporte une matrice X de dimension $500x256$ contenant 500 chiffres entre 0 et 4, et un vecteur y donnant les catégories (0, 1, 2, 3, ou 4).

Dans un premier temps nous expliquerons l'algorithme t-sne et plus finement les paramètres *perplexité* et *le nombre d'itérations* à utiliser. Puis nous comparerons cette algorithme à celle de l'ACP afin d'illustrer dans quelle situation il serait préférable d'utiliser t-sne. Enfin nous exploiterons le clusturing par k-means, dans le jeu de données digits, en utilisant le résultat du deuxième point. Ceci nous permettra de calculer les critères de pureté demandés.

algortihme t-sne

Description

t-SNE est un algorithme de réduction de dimensions non linéaire utilisé pour l'exploration de données de grande dimension. Son but est de trouver des modèles dans les données en identifiant les clusters observés sur la base de la similarité de points de données avec plusieurs caractéristiques. Ceci étant, nous ne pouvons le confondre avec un algorithme de clusturing. En effet, les données multidimensionnelles étant mappées dans un espace dimensionnel inférieur, les entités en entrée ne sont plus identifiables. Ainsi, nous ne pouvons pas faire d'inférence basée uniquement sur la sortie de t-SNE. Il s'agit donc essentiellement d'une technique d'exploration et de visualisation de données. Le but est de prendre un ensemble de points dans un espace de grande dimension et de trouver une représentation fidèle de ces points dans un espace de plus petite dimension, généralement le plan 2D. L'algorithme commence par convertir les distances euclidiennes de grande dimension entre les points de données en probabilités conditionnelles qui représentent des similitudes. La similarité entre le point x_i et le point x_j est la probabilité conditionnelle $p_{i|j}$ que x_i choisirait comme son voisin x_j si les voisins étaient choisis proportionnellement à leur densité de probabilité sous un gaussien centré sur x_i. Pour les points de données à proximité, il est relativement élevé, tandis que pour les points de données largement séparés, il sera minime.
Une autre caractéristique de t-SNE est un paramètre ajustable, la **perplexité**, qui indique comment équilibrer l'attention entre les aspects locaux et globaux de vos données. Le paramètre est en quelque sorte une estimation du nombre de voisins proches de chaque point. La valeur de perplexité a un effet complexe sur les images résultantes. Elle est définit par:

$$Perp(P_i) = 2^{H(P_i)}$$

Où

$$H(P_i) = -\sum_{j=1} p_{i|j} \log_2 p_{i|j}$$

Paramètres de t-sne

Rtsne(X, dims = 2, perplexity = 30, theta = 0.5, pca = TRUE, max_iter = 1000, verbose = getOption("verbose", FALSE),

X: représente la matrice de donnees.

dims: nombre de dimension projeté.

k: est le nombre de dimension sur laquelle on veut projetter les résultats.

initial_dims: est le nombre de dimensions à utiliser dans la méthode de réduction.

Perplexity: est le nombre optimal de voisins. Les valeurs typiques sont comprises entre 5 et 50. On observe une tendance à des formes plus claires lorsque la valeur de la préplexité augmente.

theta: vitesse de précision

pca: si TRUE, une étape de l'ACP est effectuée

max_iter: est le nombre maximum d'itérations à faire.

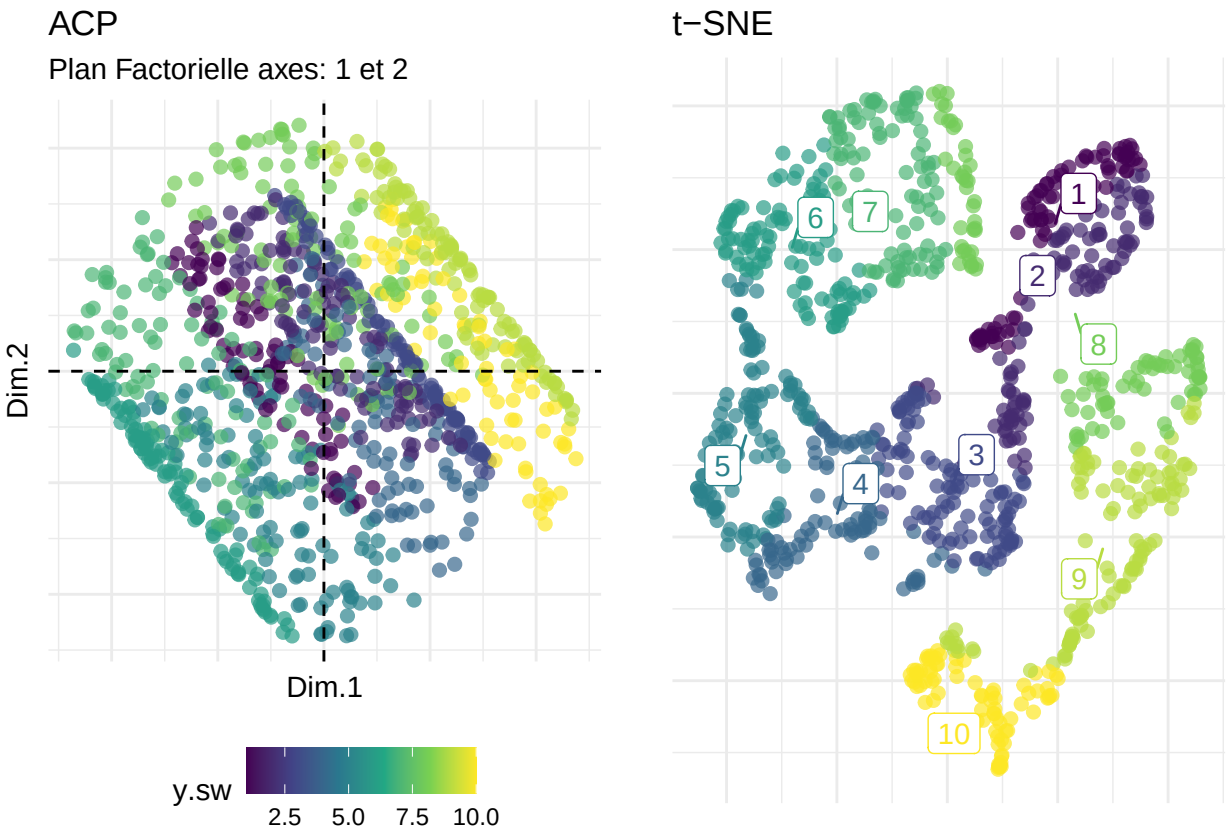
verbose: affiche les étapes et résultats de l'algorithme

Linéarité des alogrithmes

L'ACP est un algorithme linéaire. Il ne pourra pas interpréter une relation polynomiale complexe entre des caractéristiques. D'autre part, t-SNE est basé sur des distributions de probabilité avec une marche aléatoire sur des graphes de voisinage pour trouver la structure dans les données. Un problème majeur des algorithmes de réduction de dimensions linéaire est qu'ils se concentrent sur le placement de points de données dissemblables très éloignés dans une représentation de dimension inférieure. Toutefois, afin de représenter des données de haute dimension sur une variété basse, non linéaire, il est important que les points de données similaires soient représentés de manière rapprochée, ce qui n'est pas ce que font les algorithmes de réduction de dimensions linéaire. Les approches locales cherchent à cartographier les points voisins de la variété avec les points proches dans la représentation de petite dimension. Les approches globales, d'autre part, tentent de préserver la géométrie à toutes les échelles, c'est-à-dire la cartographie des points proches aux points proches et des points éloignés à des points éloignés

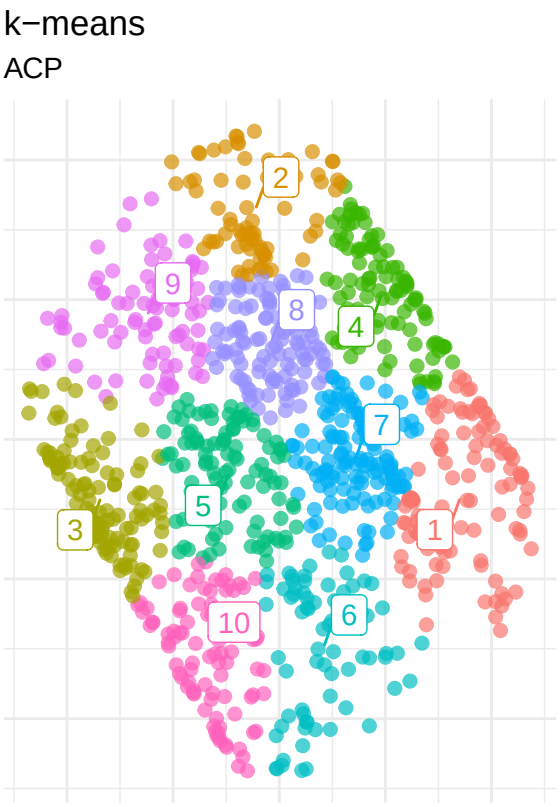
Analyse du jeu de données Swiss-Roll

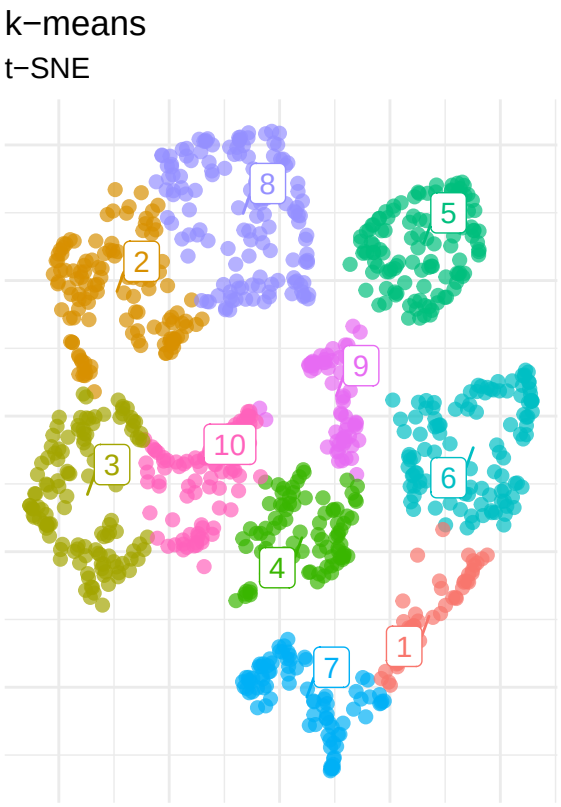
Représentation des données Swiss-Roll



Au vu des graphiques ci-dessus, il semble que t-sne regroupe de façon plus homogènes les groupes d'individus similaires. La distance intra-groupe est moins importante que celle de l'ACP où l'on voit une forme circulaire des scores. Par ailleurs, si on prend l'exemple du *point 10*, on voit bien une étalage tout au long de la deuxième bissectrice de l'ACP. Alors que sur la t-sne une nette formation d'un groupe se manifeste.

Kmeans





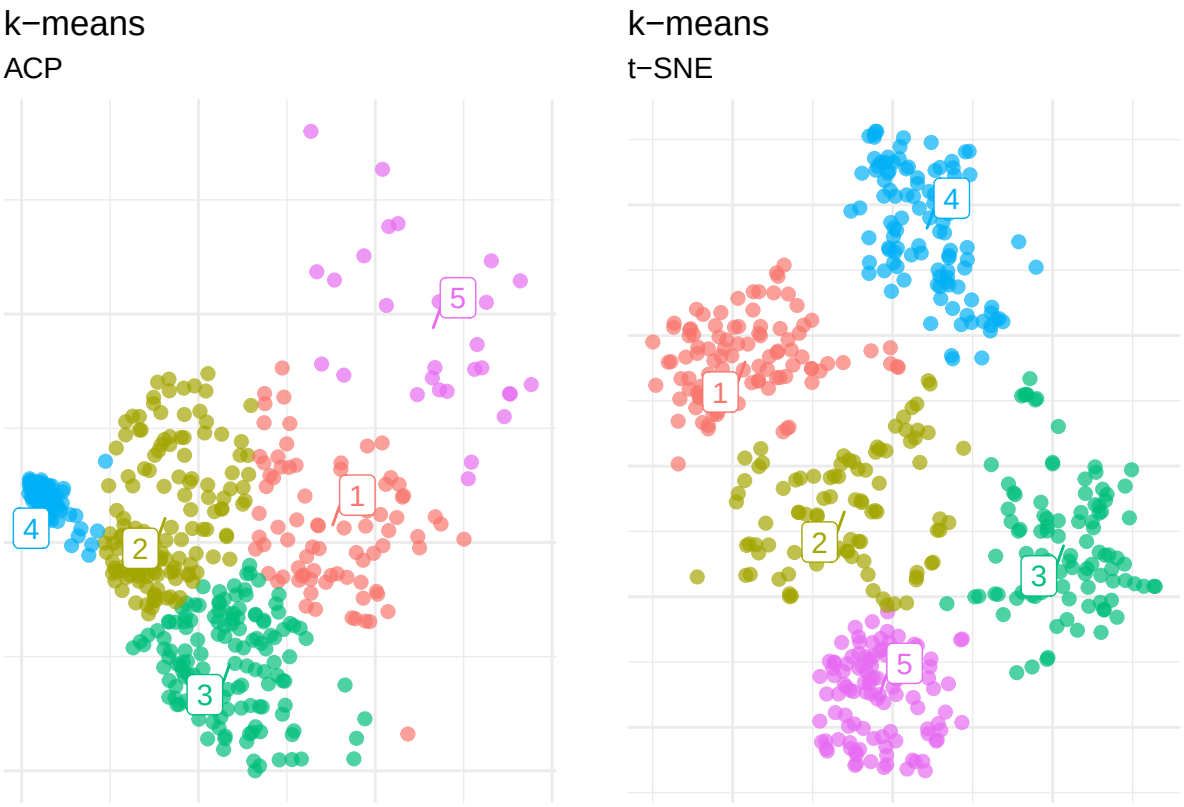
Un simple coup d'oeil sur ces graphiques montre que:
-La forme de l'ACP reste cylindrique.
-Le clusturing sur l'algorithme t-sne ressemble fortement à la représentation standard de celle-ci.
Or on sait qu'une bonne méthode de k-means doit allier une forte similarité au sein d'un même cluster et une faible similarité entre les clusters. Ce que ne remplie pas pleinement le clustering sur l'ACP. Il en résulte qu'il est plus adéquat de faire une classification sur les résultats de la t-sne.

Analyse sur le jeu de données Digits

Etudes des paramètres

Afin de déterminer la meilleure configuration possible sur les paramètres *perplexité* et *nombre d'itération*, nous avons projeté les données issues du résultat de t-sne sous différentes configurations **(voir graphiques en annexe)**.
- Nombre d'itération: on remarque que le nombre d'itérations entre 5 et 200 ne permet pas une distinction des groupes: la probabilité qu'un point x_i trouve comme voisin un point x_j est toujours élevée. Leur variance intra-groupe reste toujours faible. Cependant on s'aperçoit qu'il y'a une stabilité au bout de 300 itérations.
- Perplexité: avec une perplexité de 30, on remarque que l'algorithme reste stable malgré une augmentation de la perplexité. t-sne agit ici comme KNN, plus le nombre d'itération est petit, plus on se dirige à un sous apprentissage. A l'inverse arrivé à un certain seuil (30), il y'a une stabilité qui nous permet de choisir à priori la perplexité.

Kmeans



Il semble ici que le clusturing effectué sur la t-sne donne un meilleur résultat. Malgré une ressemblance entre les clusters, la t-sne fournit une inertie intra-groupe plus faible. Dans quatre clusters sur cinq, la distance entre les points et leur centroïde est plus petite que sur l'ACP.

TABLE 1 – Comparaison des Wss

	Cluster: 1	Cluster: 2	Cluster: 3	Cluster: 4	Cluster: 5
ACP	1312.53	1719.60	1566.69	126.08	749.82
t-SNE	518.50	788.18	818.18	758.85	435.97

Pureté dans chaque cluster

La façon la plus simple de calculer la pureté est de chercher la classe majoritaire dans chacun des clusters et de sommer le nombre d'objets de cette classe pour chacun des clusters (Manning et al., 2008). La pureté d'un clustering se définit comme :

$$\pi_{simple}(\mathbb{C}, \mathbb{W}) = \frac{1}{N} \sum_{i=1}^{n} arg \quad max(n_j^i)$$

source: Germaine Forestier

TABLE 2 – Comparaison des Puretés dans chaque Cluster

	Cluster: 1	Cluster: 2	Cluster: 3	Cluster: 4	Cluster: 5
ACP	54.32	56.43	61.31	88.50	86.21
t-SNE	92.08	93.41	90.91	94.29	96.15

La pureté dans chaque cluster est plus faible sur l'ACP que sur la t-sne. On en conclue que les clusters formés par la t-sne affiche une meilleure performance dans la construction des classes. Les individus sont plus homgènes dans chaque clusteur de la t-sne.

Pureté Globale

TABLE 3 – Comparaison des Puretés Globaux

Purete Globale ACP	Purete Globale t-SNE
69.35	93.37

La pureté globale n'étant que la moyenne des puretés des 5 clusters. Or la pureté dans chaque cluster est plus petite avec l'ACP qu'avec la t-sne. Il en résulte, sans surprise, que l'ACP détermine une pureté globale plus faible.

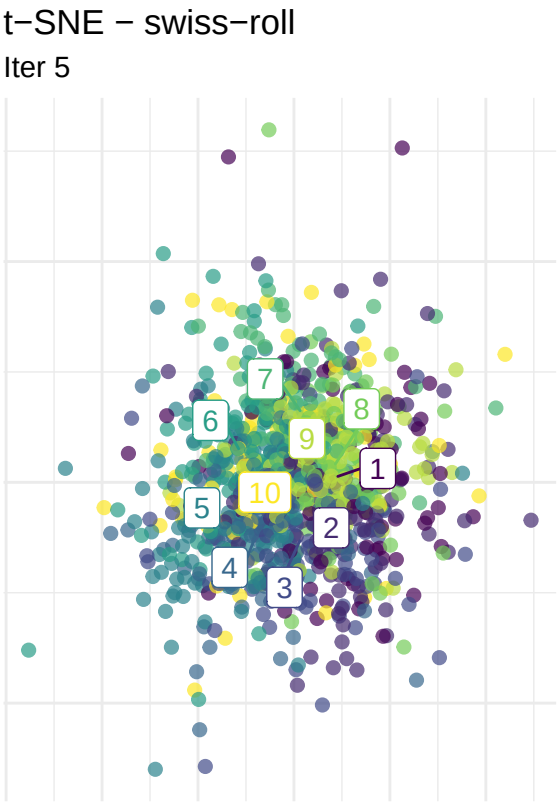
Conclusion

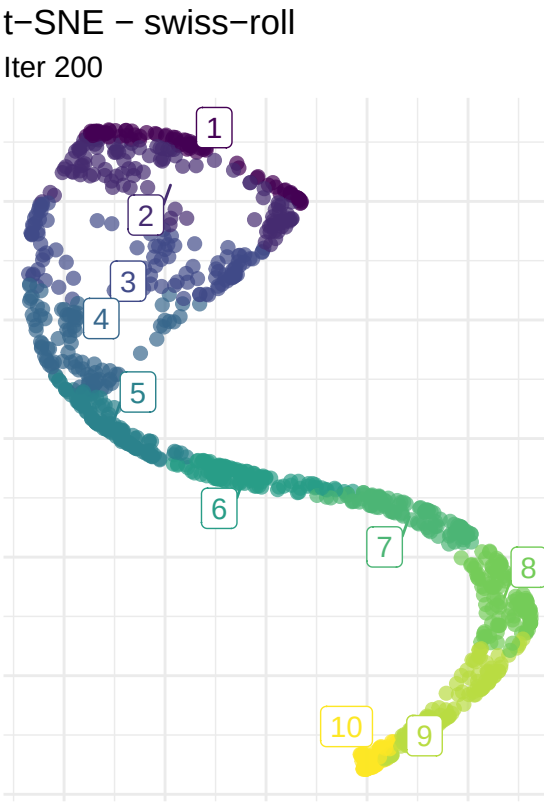
Cette étude nous a permis, d'une part, de nous familiariser avec l'algorithme t-sne. On a pu voir que sur des données non linéaires, t-sne permet d'illustrer une meilleure représentation en faisant une détection de voisins similaire à celle de l'algorithme KNN. Tandis que l'ACP, qui se base sur une calcule de distance euclidienne conserve une représentation non linéaire des données.
D'autre part le clusturing sur les données de l'ACP permet d'identifier nettement les différents groupes; que les données soient linéaires ou non. Cependant la t-sne propose une meilleure pureté dans les clusters donc une meilleure similarité entre individu.
On en conclue alors que quelque soit la linéarité des données, surtout non-linéaire, il serait préférable de réduire les dimensions avec la t-sne pour ensuite utiliser un algorithme de classification afin identifier les groupes.

Annexe

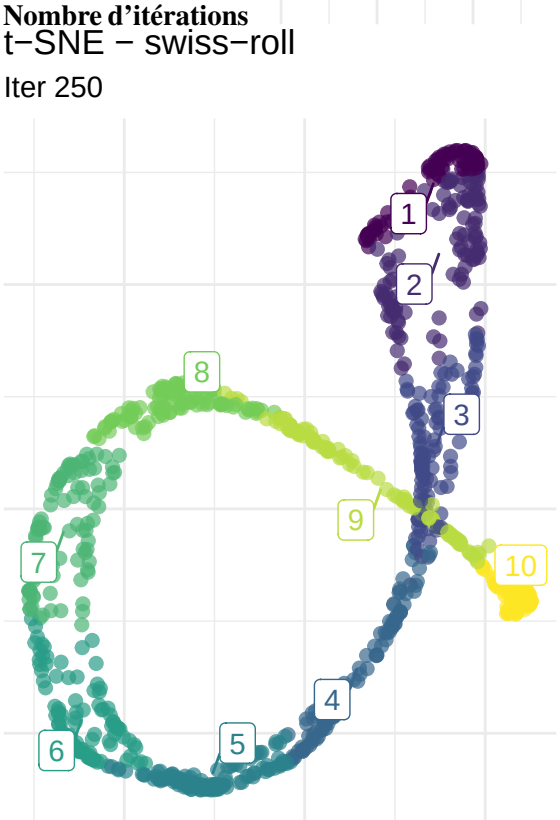
Swiss-Roll

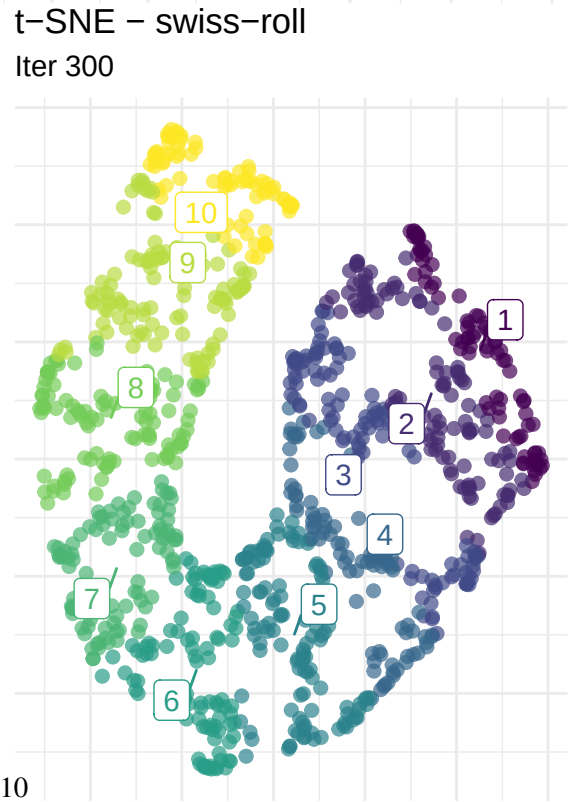
Etude des paramètres nombre d'itérations et perplexité





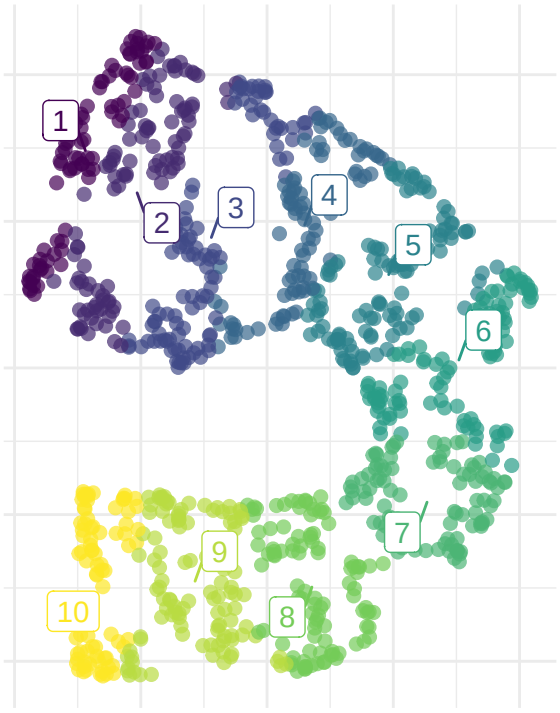
Nombre d'itérations





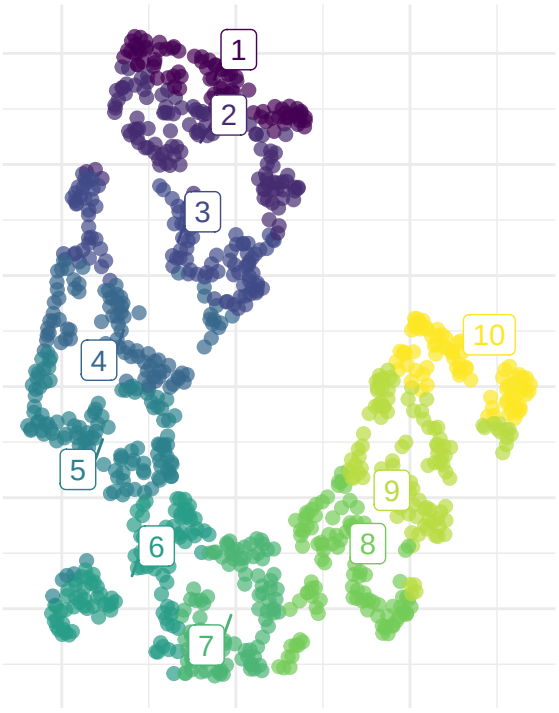
t−SNE − swiss−roll

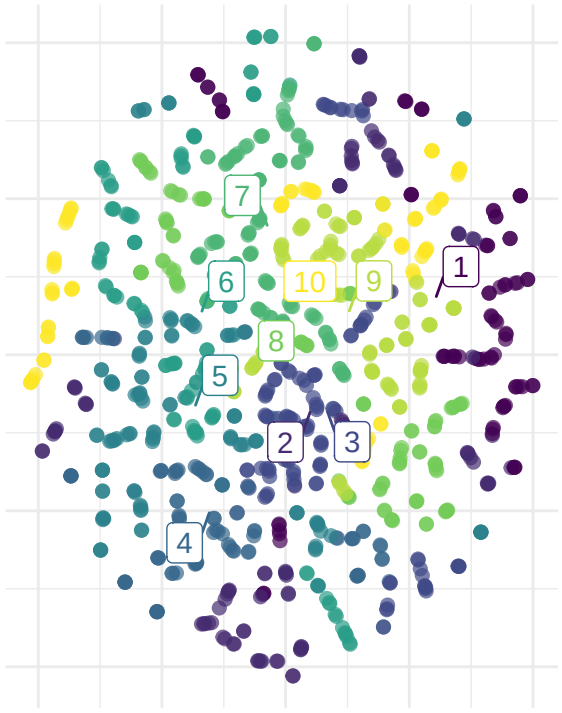
Iter 400

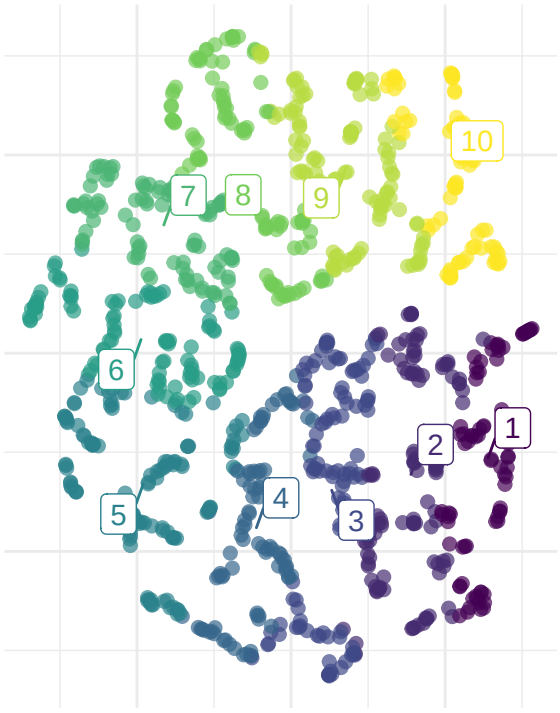


t−SNE − swiss−roll

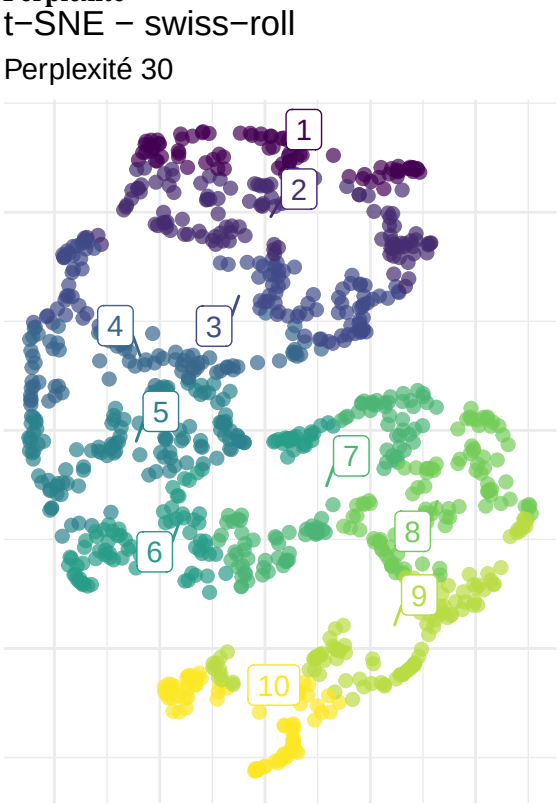
Iter 2000

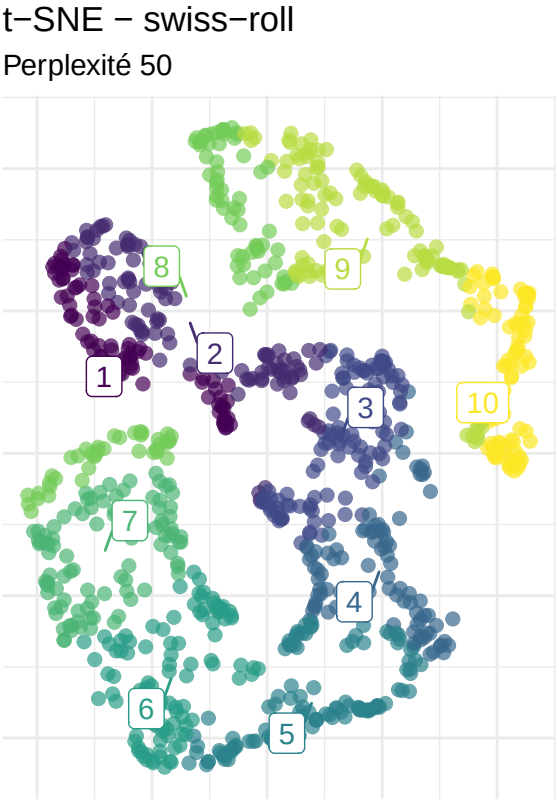



t−SNE − swiss−roll
Perplexité 2


t−SNE − swiss−roll
Perplexité 10

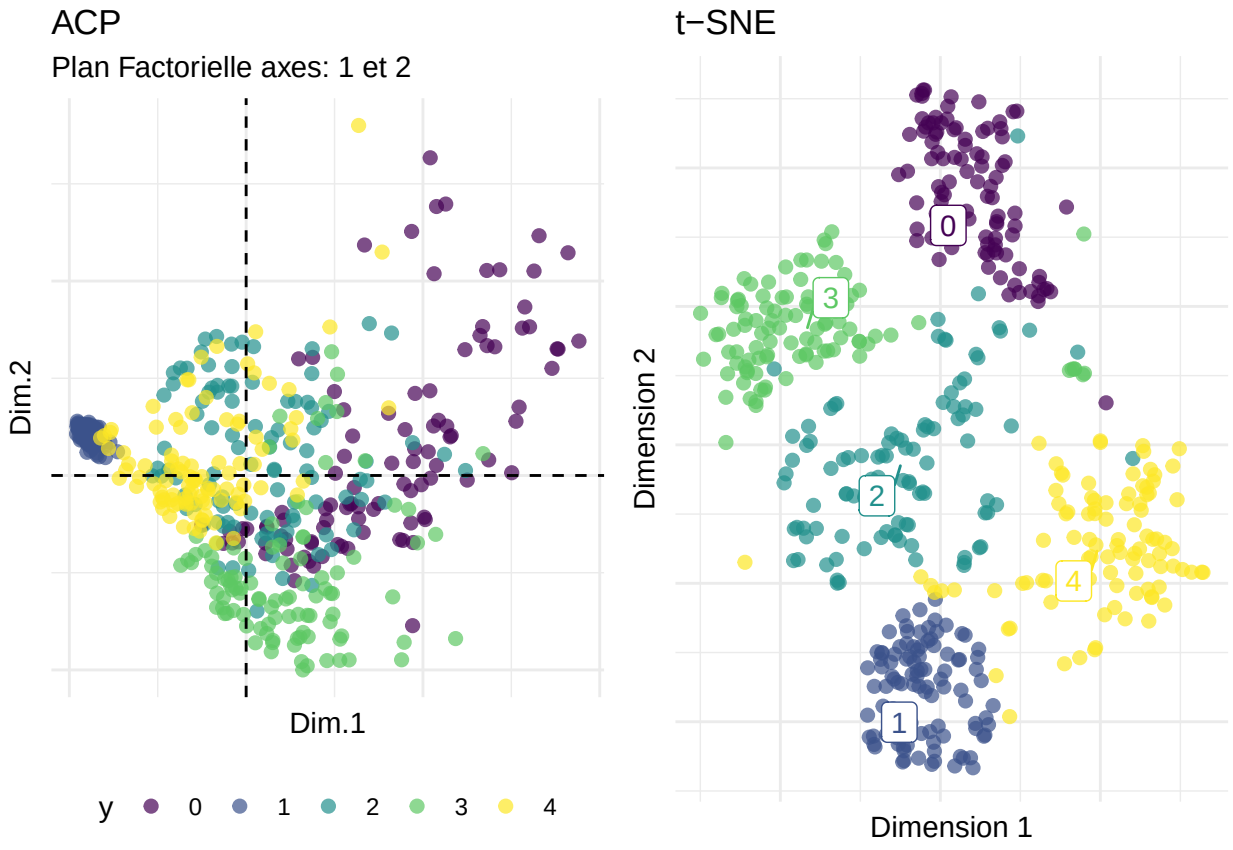
Perplexité


t−SNE − swiss−roll
Perplexité 30

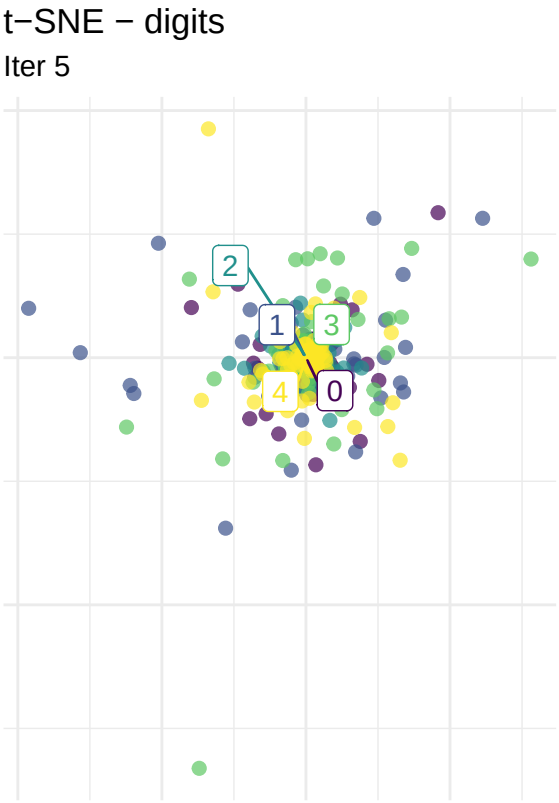

t−SNE − swiss−roll
Perplexité 50

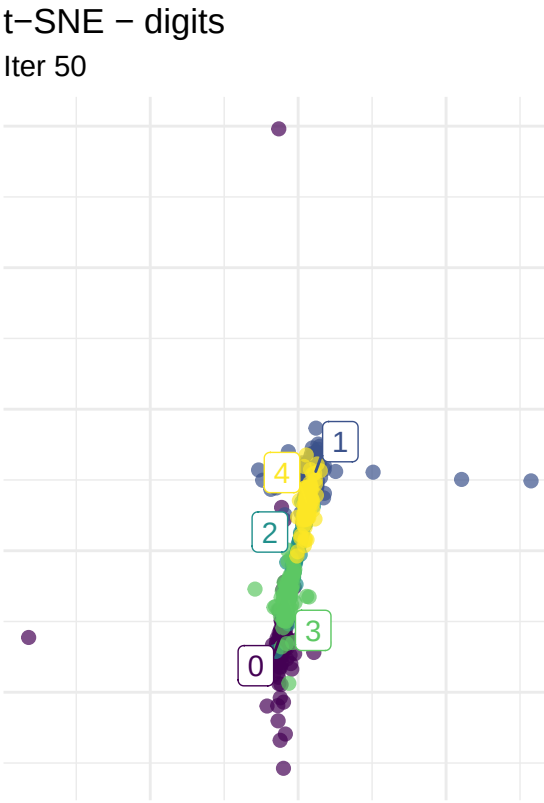
Digits:

Représentation graphique du jeu de données digits

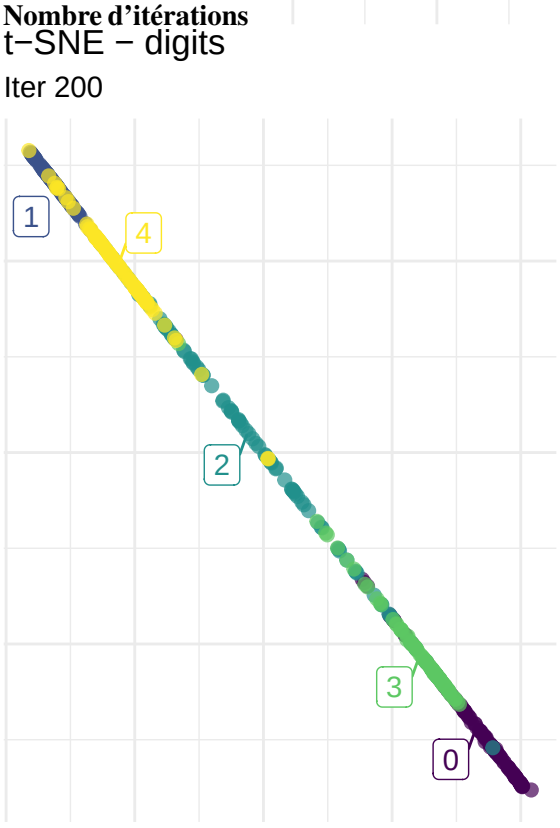


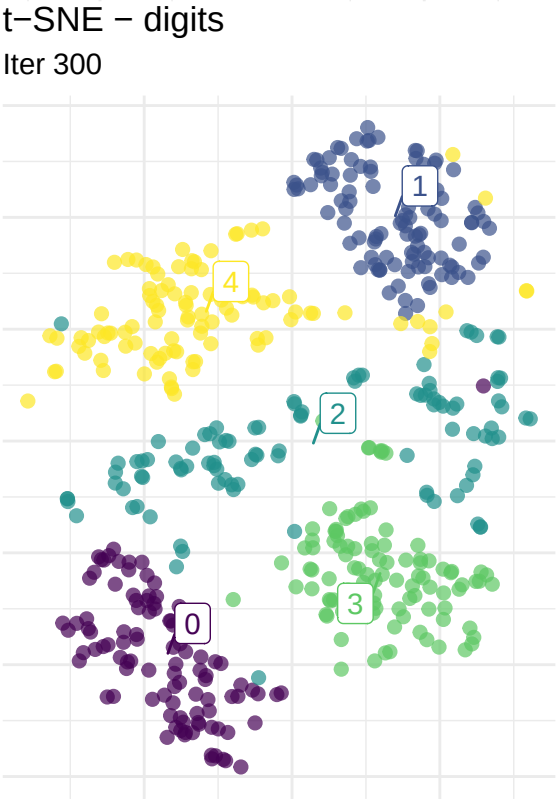
Etude des paramètres nombre d'itérations et perplexité





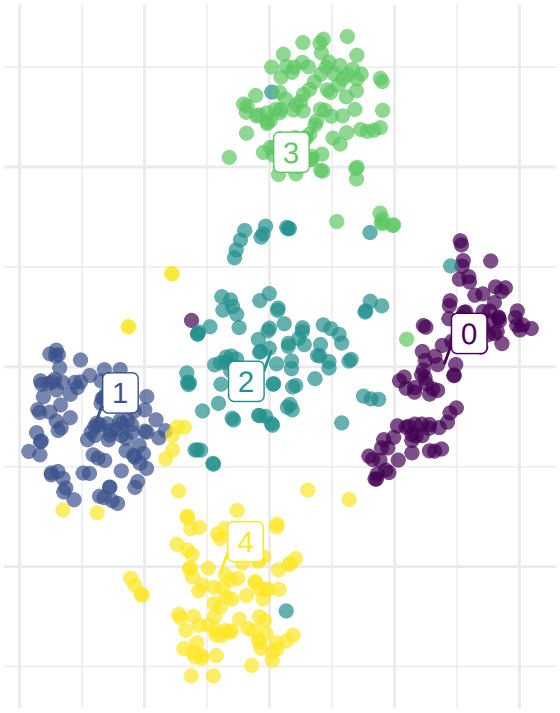
Nombre d'itérations





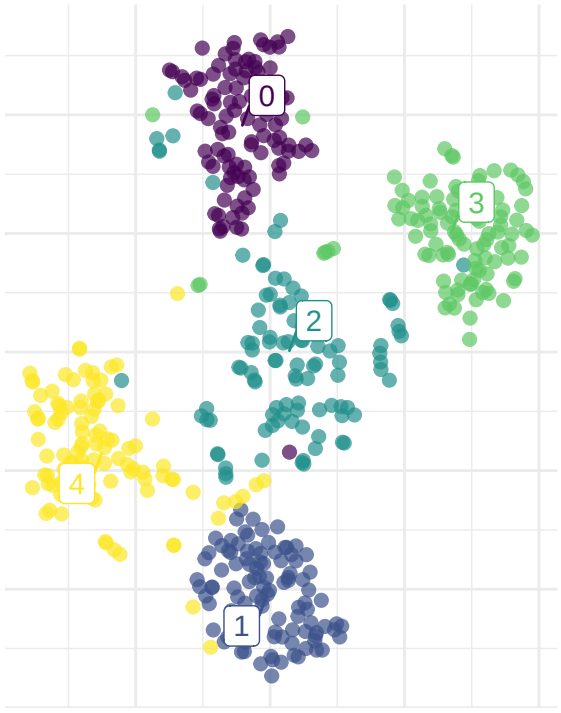
t−SNE − digits

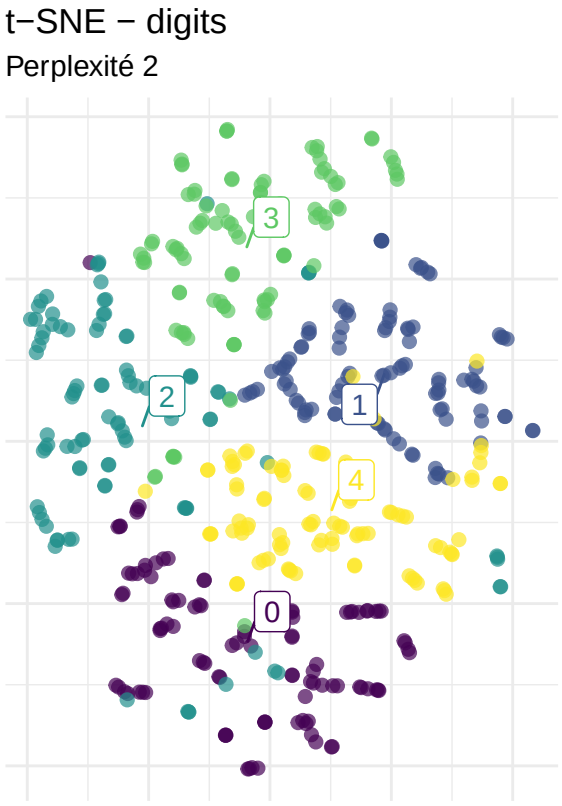
Iter 400

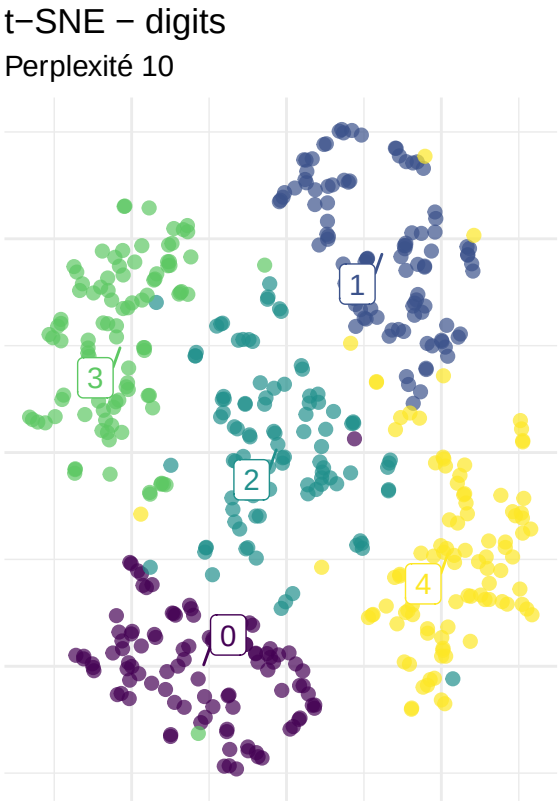


t−SNE − digits

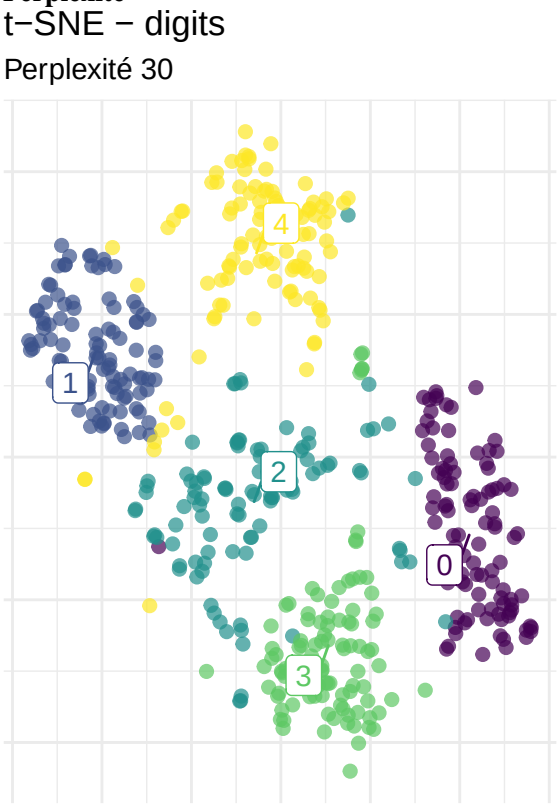
Iter 2000

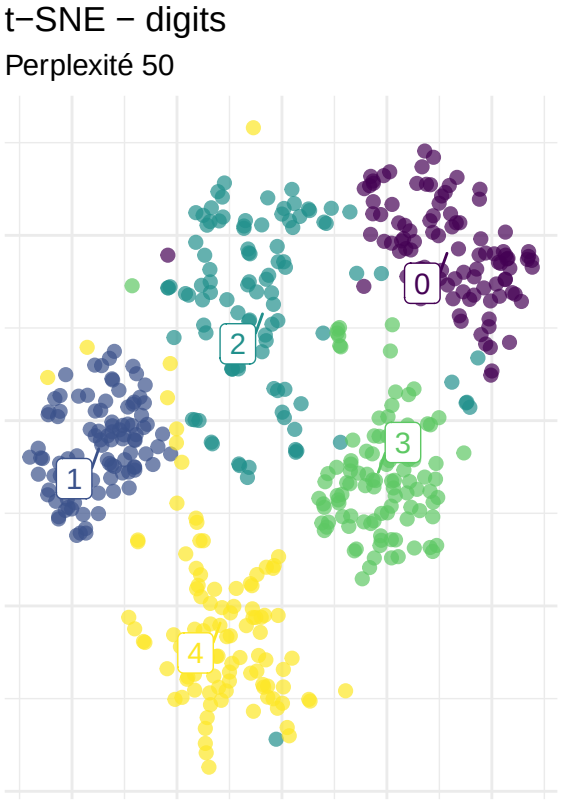



t−SNE − digits
Perplexité 2
3
2
1
4
0


t−SNE − digits
Perplexité 10
1
3
2
4
0

Perplexité


t−SNE − digits
Perplexité 30
4
1
2
0
3


t−SNE − digits
Perplexité 50
0
2
3
1
4