

Analyse, classification et indexation des données: feuille 1

Éléments de Matlab

L'objectif de ce premier TD est de rassembler les outils Matlab nécessaires dans la suite. Vous travaillerez dans le fichier `TD1.m` qui contient des instructions à tester. Vous pourrez également y ajouter les réponses aux exercices suivants. Toutes les lignes du fichier `TD1.m` sont commentées. Décommentez au fur et à mesure les instructions à tester.

Pour lancer Matlab : `$ matlab`

Documentation : <https://fr.mathworks.com/help/matlab/index.html>

1 Vecteurs et matrices

Un élément unique, un vecteur ligne ou colonne sont des cas particuliers de matrice (tableau 2D).

Exercice 1. Création de tableaux, accès à un élément d'un tableau

Testez instruction par instruction la partie **Exercice 1** du fichier `TD1.m`. Ajoutez un point virgule à la fin d'une instruction. Quelle est la différence ?

Vérifiez que vous maîtrisez les éléments de syntaxe matlab suivants :

- ☐ création d'un tableau (vecteur ligne, vecteur colonne, matrice)
- ☐ les différentes initialisations possibles (vous pouvez en lister au moins cinq)
- ☐ accès à un élément, à un sous-tableau
- ☐ duplication et concaténation de tableaux
- ☐ modification de la taille d'un tableau en conservant les mêmes éléments dans un parcours colonne par colonne

Exercice 2. Tri

1. Testez la partie **Exercice 2** du fichier `TD1.m`.
2. Créez une matrice d'entiers M de 4 lignes et 9 colonnes.
3. Triez M de telle sorte que $M(i,j) \leq M(i,k)$ pour $k > j$ et que $M(i,j) \leq M(k,l)$ avec $k > i$.
Indice : la fonction `transpose` permet d'échanger les lignes et les colonnes d'une matrice.

Exercice 3. Indexation logique

1. Testez la partie **Exercice 3** du fichier `TD1.m`.
2. Créez deux vecteurs ligne d'entiers Va et Vb , de même taille.
3. En utilisant la fonction `max` et sans utiliser de boucle, créez un nouveau vecteur $Vmax$ tel que $Vmax[i] = \max(Va[i], Vb[i])$.

- Comment obtenir le même résultat sans utiliser la fonction `max` et toujours sans utiliser de boucle ?
Indice : l'opérateur `.*` permet de calculer le produit terme à terme de deux vecteurs et on obtient la négation d'un booléen avec l'opérateur `~` (qui peut aussi s'appliquer à tout un tableau de booléens).
- Reprenez l'exercice en remplaçant le maximum par le minimum.

Exercice 4. Opérations arithmétiques sur vecteurs et matrices

Testez la partie **Exercice 4** du fichier `TD1.m`.

Vérifiez que vous maîtrisez les éléments de syntaxe matlab suivants :

- ☐ transposition
- ☐ addition/soustraction de deux vecteurs ou matrices de mêmes dimensions
- ☐ multiplication terme à terme de deux vecteurs ou matrices de mêmes dimensions
- ☐ multiplication de deux vecteurs ou matrices de dimensions compatibles

Pour vérifier ce dernier point, utilisez l'exemple suivant. Les matrices $M_{5,5}$, $M_{10,5}$, $M_{5,3}$ sont respectivement de taille 5×5 , 10×5 et 5×3 . Le vecteur V_{L5} est un vecteur ligne de taille 5 et le vecteur V_{C5} un vecteur colonne de taille 5. Quels sont les produits possibles ?

*	$M_{5,5}$	$M_{10,5}$	$M_{5,3}$	V_{L5}	V_{C5}
$M_{5,5}$					
$M_{10,5}$					
$M_{5,3}$					
V_{L5}					
V_{C5}					

2 Affichage d'ensembles de points, définition de fonctions

Exercice 5. Tests en 2D et 3D

Testez la partie **Exercice 5** du fichier `TD1.m`.

Cherchez dans la documentation quelle fonction mathématique correspond à la fonction Matlab `normpdf`. Expérimentez différentes façons de paramétrer `plot` (couleur et style de la ligne, marquage des points). Quelle est la différence entre `plot` et `scatter` ?

Exercice 6. Affichage de droites en 2D

- Ecrire la fonction `y = droite2DVD (x, vd, p)` telle que le point (x,y) appartienne à la droite passant par le point `p` et dont le vecteur directeur est `vd`. Afficher la droite définie par `vd = [1 2]` et `p = [1 1]`.
- Ecrire la fonction `y = droite2DNorm (x, n, p)` telle que le point (x,y) appartienne à la droite passant par le point `p` et dont la normale est donnée par le vecteur `n`. Afficher la droite définie par `n = [1 2]` et `p = [1 1]`.

3 Application : régression linéaire

On suppose qu'on dispose d'un ensemble d'observations composé de points 2D :

$$\text{Obs} = \{(x_1, y_1), \dots, (x_m, y_m)\}$$

On suppose également que la relation entre x et y est linéaire : $y = \theta_1 x + \theta_0$. Le code matlab suivant permet de générer un ensemble **Obs** en bruitant les points d'une droite :

```
m = 40;
sizeNoise = 10;
x = rand(m,1).*50 + 5;
noise = rand(m,1) * sizeNoise;
pente = 0.8;
c = 20;
y = c + pente*x + noise;
```

1. Affichez le nuage de points correspondant.
2. La droite de régression linéaire basée sur les moindres carrés peut être calculée avec :

$$\theta_1 = \frac{m \sum (x_i y_i) - \sum (x_i) \sum (y_i)}{m \sum (x_i^2) - (\sum (x_i))^2}$$

$$\theta_0 = \frac{\sum (y_i) - \theta_1 \sum (x_i)}{m}$$

Affichez cette droite sur le nuage de points.

3. Comparer votre résultat avec celui de la fonction matlab **regress**.