

Enseignement annuel d'informatique et d'algorithmique

I - Éléments d'informatique et d'algorithmique

En première année, les élèves ont consolidé les bases de manipulation du langage Python. L'objectif de l'enseignement d'informatique de seconde année est de permettre aux étudiants de l'utiliser de manière judicieuse et autonome pour illustrer ou modéliser des situations concrètes en mobilisant leurs connaissances mathématiques.

Le programme d'informatique s'articule autour de quatre thèmes : statistiques descriptives bivariées, études de suites et de fonctions, simulation de lois, estimation.

L'ordre dans lequel les thèmes sont abordés est libre, mais il est préférable de mener ces activités en cohérence avec la progression du cours de mathématiques.

Les exemples traités dans un thème devront être tirés, autant que possible, de situations réelles (traitement de données économiques, sociologiques, historiques, démographiques, en lien avec le monde de l'entreprise ou de la finance, etc.), en faisant dès que possible un rapprochement avec les autres disciplines.

Pour certains thèmes, il sera nécessaire d'introduire de nouvelles notions mathématiques ; celles-ci seront introduites lors des séances d'informatique ; elles ne pourront en aucun cas être exigibles des étudiants, et toutes les précisions nécessaires seront données lors de leur utilisation.

Le langage informatique retenu pour la programmation dans ce programme des classes économiques et commerciales, option technologique, est Python.

Toute la richesse du langage Python ne peut pas être entièrement maîtrisée par un étudiant, aussi seules les fonctions et commandes exigibles du programme de première année sont exigibles, et leur syntaxe précise doit être rappelée. D'autres fonctions, par commodité, pourront être utilisées en classe, mais ceci ne pourra se faire qu'avec parcimonie. L'objectif principal de l'activité informatique reste la mise en pratique de connaissances mathématiques. Ces commandes supplémentaires devront être présentées en préambule et toutes les précisions nécessaires devront être données lors de leur utilisation et leur interprétation. On favorisera à cette occasion l'autonomie et la prise d'initiatives des étudiants grâce à l'utilisation de l'aide de Python, et à l'usage d'opérations de «copier-coller» qui permettent de prendre en main rapidement des fonctions nouvelles et évitent d'avoir à connaître par cœur la syntaxe de commandes complexes.

L'objectif de ces travaux pratiques n'est pas l'écriture de longs programmes mais l'assimilation de savoir-faire et de compétences spécifiés dans la liste des exigibles et rappelés en préambule de chaque thème.

1 - Liste des savoir-faire et compétences

C1 : Produire et interpréter des résumés numériques et graphiques d'une série statistique (simple, double) ou d'une loi.

C2 : Modéliser et simuler des phénomènes (aléatoires ou déterministes) et les traduire en langage mathématique.

C3 : Porter un regard critique sur les méthodes d'estimation et de simulation.

C4 Stocker, organiser et extraire des données structurées volumineuses.

II - Langage Python

Les commandes exigibles ont été listées dans le programme de première année.

III - Liste des thèmes

1 - Statistiques descriptives bivariées

(Durée indicative : 4 heures. Compétences développées : **C1** et **C3**)

On s'appuiera sur les représentations graphiques pour montrer l'intérêt et les limites des indicateurs.

Série statistique à deux variables, nuage de points associé.

Point moyen (\bar{x}, \bar{y}) du nuage.

Covariance empirique, coefficient de corrélation empirique, droites de régression.

On tracera le nuage de points et on pourra effectuer des pré-transformations pour se ramener au cas linéaire.

Analyse de deux caractères quantitatifs : covariance empirique, corrélation linéaire empirique, ajustement affine par la méthode des moindres carrés.

On différenciera les variables explicatives des variables à expliquer et on soulignera la distinction entre corrélation et causalité.

On pourra donner des exemples d'utilisation de la droite de régression pour faire des prévisions dans le cadre de problèmes concrets.

On pourra utiliser les commandes : `plt.scatter`, `np.polyfit`, `np.corrcoef` ou un tableur

2 - Simulation de lois, application au calcul d'espérances

(Durée indicative : 4 heures. Compétences développées : **C1**, **C2**, et **C3**)

Ces simulations de variables aléatoires seront introduites comme illustrations de problèmes concrets, et permettront d'en vérifier la compréhension par les étudiants. Dans toutes les simulations effectuées, on pourra comparer les échantillons obtenus avec les distributions théoriques, en utilisant des diagrammes en bâtons et des histogrammes. On pourra aussi tracer la fonction de répartition empirique et la comparer à la fonction de répartition théorique. On pourra utiliser les générateurs de nombres aléatoires selon les lois uniformes, binomiales, géométriques, normales, de la bibliothèque `numpy.random` : `rd.random`, `rd.binomial`, `rd.randint`, `rd.geometric`, `rd.poisson`, `rd.exponential`, `rd.normal` .

Simulation de la loi uniforme sur $[0, 1]$; sur $[a, b]$.

Simulation de phénomènes aléatoires à partir de lois usuelles.

Méthodes de simulation d'une loi géométrique.

Comparaison entre différentes méthodes : utilisation d'une loi de Bernoulli et d'une boucle `while`, utilisation du générateur `rd.random`.

Simulation de lois usuelles.

3 - Bases de données

(Durée indicative : 4 heures. Compétences développées : **C4**)

Dans la continuité du programme de première année, on poursuit l'étude du langage SQL avec la création de table et l'interrogation avancée via l'instruction JOIN. On introduira ces concepts à l'aide d'exemples simples issus de contextes appropriés.

a) Commandes exigibles

"SELECT* FROM nom_de_table_1 INNER JOIN nom_de_table_2".

Réalisation d'une jointure. On pourra ajouter une condition "ON Φ " dans le cas où Φ est une conjonction d'égalités.

Aucune autre notion de jointure n'est dans ce programme.

"CREATE TABLE nom_de_table".

Création d'une table.

b) Commandes non exigibles

Les commandes non exigibles ont été listées dans le programme de première année.

4 - Théorème limite central

(Durée indicative : 4 heures. Compétences développées : **C1**, **C2**, et **C3**)

L'objectif est ici de dégager des conséquences importantes du théorème limite central qui n'est pas au programme. On met en œuvre sur des exemples ce théorème, qu'on pourra énoncer sans formaliser la notion de convergence. On souhaite dégager la pertinence de l'utilisation de la loi normale pour modéliser les phénomènes résultant de nombreux phénomènes aléatoires indépendants et de l'intervalle de confiance asymptotique dont on pourra mettre en valeur la précision.

Etude de la distribution des moyennes empiriques par simulation informatique de la loi de $X = \frac{Y_1 + \dots + Y_n}{n}$ où Y_1, \dots, Y_n sont des variables aléatoires indépendantes suivant toutes une même loi discrète d'espérance μ et d'écart type σ .

On cherche à visualiser la convergence vers la loi normale d'espérance μ et d'écart type σ . On pourra produire un échantillon de taille N des moyennes empiriques d'échantillons de taille n d'une variable aléatoire d'espérance μ et d'écart type σ , et le représenter sous forme d'un histogramme. On observera l'effet de l'augmentation de n sur la dispersion des moyennes.

Simulation informatique de la loi de $X = \frac{Y_1 + \dots + Y_n}{n}$ où Y_1, \dots, Y_n sont des variables aléatoires indépendantes suivant toutes la loi uniforme à densité sur $[0, 1]$.

On remarquera que la variable aléatoire centrée réduite associée à X est une approximation de la loi normale centrée réduite et on sensibilisera les étudiants au théorème limite central, en testant cette simulation avec d'autres lois.

Intervalle de confiance asymptotique.

On compare l'intervalle de confiance obtenu avec l'inégalité de Bienaymé-Tchebychev avec l'intervalle de confiance asymptotique, qu'on présentera en invoquant le théorème limite central pour estimer le paramètre d'une loi de Bernoulli.