

PROGRAMME DE FORMATION

LIBRAIRIES DE DATA SCIENCE AVEC PYTHON

FICHE SYNTHÉTIQUE DE LA FORMATION

Intitulé	Librairies de Data Science avec Python —
Durée	4 jours (28 heures)
Modalités	Présentiel ou distanciel synchrone (classe virtuelle)
Certification visée	RS6701 — Manipuler, analyser et visualiser des données grâce aux modules Python de Data Science (API Society)
Tarif	Sur devis (inter / intra)
Délais d'accès	Inscription possible jusqu'à 14 jours avant le début de la session
Lieu	Paris 17e ou distanciel via Zoom
Effectif	4 participants maximum par session
Contact	contact@pythonia.fr – 06 66 06 38 59
Accessibilité PSH	Formation accessible aux PSH. Contactez notre référent handicap pour toute adaptation.
Dernière mise à jour	Février 2026

1. Public visé

Cette formation s'adresse aux professionnels ayant des bases en Python souhaitant acquérir les compétences de Data Science appliquées à la finance de marché : analystes quantitatifs juniors, risk managers, data analysts en banque ou société de gestion, développeurs souhaitant évoluer vers la finance quantitative.

2. Prérequis

- Maîtriser les bases de Python (variables, types, boucles, conditions, fonctions, fichiers)
- Notions de base en mathématiques financières appréciées (non obligatoire)

Un test de positionnement est proposé avant l'entrée en formation afin de vérifier l'adéquation du parcours avec le profil du participant.

3. Objectifs pédagogiques

À l'issue de cette formation, le stagiaire sera capable de :

- Manipuler des jeux de données financières avec NumPy et Pandas (chargement, nettoyage, agrégation)
- Créer des visualisations de données professionnelles avec Matplotlib, Seaborn et Plotly
- Analyser des séries temporelles boursières et calculer des indicateurs financiers
- Produire des tableaux de bord interactifs orientés finance de marché

4. Certification préparée

Code RS	RS6701
---------	--------

Intitulé	Manipuler, analyser et visualiser des données grâce aux modules Python de Data Science
Certificateur	API Society
Équivalences	Aucune équivalence identifiée à ce jour
Passerelles	Complète des parcours en data science, finance quantitative, IA
Débouchés	Analyste quantitatif, data analyst finance, développeur Python finance

5. Programme détaillé

Toutes les démonstrations et travaux pratiques s'appuient sur des données réelles de finance de marché (cours boursiers, indices, taux, volumes, portefeuilles d'actifs).

JOUR 1 — L'écosystème Python scientifique & NumPy (7h)

Matin (3h30) — Écosystème Python scientifique & Introduction NumPy

L'écosystème Python scientifique

- Tour d'horizon des packages Python de Data Science
- Installation de bibliothèques scientifiques dans un environnement virtuel : pip et le module venv, miniconda, mamba, miniforge, WinPython
- Environnement de développement : IPython, Jupyter Notebook, JupyterLab, IDE (Spyder), éditeur de texte (VS Code)

Introduction à NumPy

- Présentation de la librairie NumPy
- Intérêts des tableaux : performance, représentation des données (matrices, séries de temps, images, données géographiques, données génériques) et facilité de manipulation
- Création de tableaux : fonctions `array()`, `zeros()`, `ones()`, `full()`, `arange()`, `linspace()`, `logspace()`
- Multiplication matricielle avec `np.dot` et l'opérateur `@`
- Créer une matrice identité avec `identity()` et `eye()`
- Construire une matrice diagonale avec `diag()`
- Initialisation avec des données aléatoires (module `random` de NumPy)
- Types de données (`bool`, `int`, `uint`, `float`, `complex`, `unicode`) et changer le type avec `astype()`
- Attributs `ndim`, `shape`, `size`, `dtype`, `itemsize`, `nbytes`

Fil rouge finance :

- Représentation matricielle d'un portefeuille d'actifs (poids, rendements)
- Génération de rendements aléatoires pour simulation Monte Carlo

Après-midi (3h30) — Manipulation de tableaux & opérations

Manipulation de tableaux

- Indexation, slicing, indexation avancée
- Broadcasting pour l'indexation et les opérations sur les données
- Copie et vue d'un tableau
- Transposer un tableau : méthode `transpose()` ou attribut `T`
- Changer les dimensions : fonctions `reshape()` et `newaxis()`
- Concaténer des tableaux : `concatenate()`, `vstack()`, `hstack()`, `stack()`
- Découper des tableaux : `split()`, `hsplit()`, `vsplit()`

Opérations et fonctions

- Pourquoi éviter les boucles `for` ?
- Fonctions classiques (addition, soustraction...), trigonométriques, exposants et logarithmes
- Fonctions `sum()`, `min()`, `max()`, `median()`, `percentile()`, `prod()`, `cumsum()`, `var()`, `ravel()`, `argmin()`, `argmax()`, `any()`, `all()`, `where()`
- Utiliser l'option `axis`
- Fonctions de comparaisons
- Masques booléens pour extraire des informations
- Charger et sauvegarder les tableaux : `loadtxt()` (options `usecols`, `skiprows`), `save()`, `load()`

Fil rouge finance :

- Calcul vectorisé de rendements journaliers et de volatilité historique
- Matrice de variance-covariance d'un portefeuille multi-actifs

Travaux pratiques — Jour 1

- TP1 : Création et manipulation de matrices de rendements boursiers (CAC 40, S&P 500)
- TP2 : Calcul de la volatilité annualisée et du ratio de Sharpe d'un portefeuille
- TP3 : Simulation Monte Carlo de l'évolution d'un actif financier

✓ *Évaluation formative : correction collective des TP, tour de table de fin de journée*

JOUR 2 — Manipulation de données avec Pandas (7h)

Matin (3h30) — Introduction à Pandas & structures de données

Présentation et structures

- Présentation de la librairie Pandas
- Créer une série avec la classe Series
- Créer un DataFrame avec la classe DataFrame
- Extraire les indices de ligne et de colonnes (attributs index et columns)
- Lire et exporter des données dans différents formats (CSV, Excel...)
- Méthodes head(), tail()
- Exploration de données : info(), describe(), dtypes

Indexation et sélection

- Indexation implicite et explicite
- Utilisation des indexeurs loc et iloc
- Sélection avancée : expressions booléennes, méthode query()
- Insérer et modifier des données
- Renommer une colonne avec rename()
- Concaténer des données avec concat()
- Fusion et jointure : merge() et join()
- Copie superficielle ou profonde : copy()

Fil rouge finance :

- Chargement de données boursières multi-actifs depuis des fichiers CSV
- Jointure de données de prix avec des données de volumes et de dividendes

Après-midi (3h30) — Nettoyage, agrégation & fonctionnalités avancées

Nettoyage de données

- Valeurs manquantes : isna(), isnull(), notna(), notnull(), dropna(), fillna(), interpolate()
- Mettre une colonne en indice : set_index()
- Trier les indices : sort_index()
- Trier les valeurs : sort_values()
- Transposer des données : transpose()
- Supprimer des données : drop()
- Supprimer les doublons : drop_duplicates()

Agrégation et analyse

- Fonctions d'agrégation : sum(), cumsum(), min(), max(), count(), mean(), median(), var(), std(), quantile(), describe()
- Grouper et analyser : groupby()
- Fonctions aggregate(), apply(), filter(), transform()
- Tableaux croisés dynamiques : pivot_table()
- Segmenter les données : qcut() et cut()
- Moyennes glissantes : rolling(), expanding(), ewm()

Multi-indices

- Créer des multi-indices : MultiIndex.from_product(), from_tuple(), from_arrays()
- Ajouter des dimensions aux Series et DataFrames
- Indexer des données multi-indexées avec pd.IndexSlice

Chaînes de caractères

- Indexation, slicing et méthodes de chaînes dans Pandas
- Enchaîner plusieurs méthodes de chaînes

- Expressions régulières avec Pandas : `match()`, `extract()`, `findall()`, `replace()`, `contains()`, `count()`, `split()`, `rsplit()`

Données temporelles

- Créer dates, durées et périodes : `to_datetime()`, `to_timedelta()`, `date_range()`, `period_range()`, `timedelta_range()`
- Indexation et slicing de données temporelles
- Échantillonnage : `asfreq()` et `resample()`

Fil rouge finance :

- Nettoyage d'un historique de cours (jours fériés, valeurs aberrantes, données manquantes)
- Calcul de moyennes mobiles (SMA, EMA) et de rendements cumulés
- Rééchantillonnage de données intra-day vers des fréquences journalières/mensuelles

Travaux pratiques — Jour 2

- TP1 : Nettoyage et exploration d'un historique boursier (données CAC 40 sur 5 ans)
- TP2 : Analyse sectorielle avec `groupby` (performance par secteur, volatilité par industrie)
- TP3 : Rapport d'analyse de portefeuille avec agrégations, moyennes mobiles et export Excel

✓ *Évaluation formative : QCM mi-parcours (20 questions), correction des TP en groupe*

JOUR 3 — Visualisation avec Matplotlib & Seaborn (7h)

Matin (3h30) — Matplotlib

Fondamentaux Matplotlib

- Présentation de la librairie
- Afficher les graphiques depuis un script Python (`plt.show()`) ou depuis un notebook
- Style MATLAB vs style orienté objet
- Modifier le style du graphique
- Les objets Figure et Axes
- Tracer des courbes : `plot()` — couleur, style, largeur, axes, titre, axes nommés, graduations, légende
- Nuages de points : `scatter()`
- Barres d'erreurs : `errorbar()`
- Remplissage entre courbes : `fill_between()`
- Histogrammes : `hist()`

Graphiques multiples et 3D

- Placer un graphique dans un autre : `add_axes()`
- Créer une grille : `subplots()`
- Graphiques en 3 dimensions avec `mplot3d`
- Widget interactif pour graphiques interactifs dans Jupyter
- Pandas plot : tour d'horizon des méthodes `plot()`, `bar()`, `barh()`, `hist()`, `box()`, `scatter()`, `pie()`
- Conseils pour améliorer la lisibilité pour les personnes en situation de handicap visuel

Fil rouge finance :

- Tracé de courbes de prix avec moyennes mobiles superposées
- Histogramme de distribution des rendements journaliers
- Graphique `fill_between` pour les bandes de Bollinger

Après-midi (3h30) — Seaborn

La bibliothèque Seaborn

- Présentation de la librairie
- API Seaborn : Figure-level et Axes-level
- Relational plots : `relplot()`, `lineplot()`, `scatterplot()`
- Distributions : `displot()`, `histplot()`, `jointplot()`, `pairplot()`
- Données qualitatives : `catplot()`, `barplot()`, `countplot()`, `boxplot()`, `violinplot()`, `stripplot()`, `swarmplot()`
- Cartes thermiques : `heatmap()`
- Modèles de régression linéaire : `lmplot()`
- Personnalisation : `set_theme()`, `set_style()`, `set_context()`, `despine()`

Fil rouge finance :

- Heatmap de la matrice de corrélation entre actifs d'un portefeuille
- Pairplot des rendements sectoriels
- Boxplot de la volatilité par secteur d'activité

Travaux pratiques — Jour 3

- TP1 : Dashboard statique multi-graphiques d'un actif (prix, volume, rendements, distribution)
- TP2 : Heatmap de corrélation et analyse visuelle d'un portefeuille diversifié
- TP3 : Visualisation complète d'un rapport de performance : bandes de Bollinger, drawdown, rendements cumulés

✓ *Évaluation formative : correction collective des TP, tour de table de fin de journée*

JOUR 4 — Visualisation interactive avec Plotly (7h)

Matin (3h30) — Plotly Express : graphiques fondamentaux

Introduction à Plotly

- Présentation de la librairie Plotly et de Kaleido
- Exploration de Plotly Express

Courbes et aires

- Tracer des courbes avec `line()` : options `title`, `width`, `height`, `marker`, `labels`
- Sauvegarder la figure
- Tracer plusieurs courbes grâce à l'option `color`
- Ajout d'informations : `hover_data`, `hover_name`, `text`
- Graphiques multiples : `facet_row`, `facet_col`
- Barres d'erreurs : `error_x`, `error_y`
- Modifier le style : option `template`, thème par défaut avec `plotly.io.templates.default`
- Graphiques en aires : `area()` avec motifs `pattern_shape`

Nuages de points

- `scatter()` : options `size`, `size_max`, `opacity`, `symbol`, `symbol_sequence`
- Barre de couleur : `color_continuous_scale`
- `update_layout()` pour modifier la position de la légende
- `update_coloraxes()` pour modifier la position de la barre de couleurs

Fil rouge finance :

- Courbes interactives de prix multi-actifs avec hover détaillé
- Scatter rendement vs volatilité avec taille proportionnelle à la capitalisation

Après-midi (3h30) — Plotly Express : graphiques avancés & cartes

Diagrammes en barres et histogrammes

- Diagrammes en barres : `bar()`
- Histogrammes : `histogram()`

Graphiques 3D

- `scatter_3d()` et `line_3d()`

Cartes géographiques

- Cartes interactives : `line_map()` et `scatter_map()` avec options `zoom` et `center`
- Cartes géographiques : `line_geo()` et `scatter_geo()` avec `scope` et `projection`
- Carte choroplèthe : `choropleth()`

Fil rouge finance :

- Surface 3D rendement/risque/corrélation d'un portefeuille
- Carte choroplèthe de performance boursière par pays/région
- Dashboard interactif complet de suivi de portefeuille

Travaux pratiques — Jour 4

- TP1 : Graphique interactif de frontière efficiente (rendement/risque) avec Plotly scatter
- TP2 : Carte choroplèthe de la répartition géographique des investissements
- TP3 : Dashboard interactif complet de suivi de portefeuille (prix, rendements, allocation, indicateurs clés)

✓ *Évaluation formative : QCM final de 30 questions — critère de réussite : 60 %*

6. Méthodes et moyens pédagogiques

Méthodes mobilisées

- Apports théoriques : exposés interactifs avec diaporama
- Mise en pratique : TP individuels, exercices progressifs sur données financières réelles
- Pédagogie active : résolution de problèmes collaboratifs
- Alternance théorie/pratique : ratio 30 % théorie / 70 % pratique

Moyens techniques

- Salle de formation équipée : vidéoprojecteur, paperboard, connexion Wi-Fi haut débit
- Poste informatique par stagiaire avec VS Code
- En distanciel : Zoom avec partage d'écran, tableau blanc collaboratif
- Supports de cours remis aux stagiaires
- Jeux de données financiers fournis (historiques de cours, indices, taux)

7. Modalités d'évaluation

Positionnement à l'entrée

Avant l'entrée en formation, chaque participant complète un test de positionnement évaluant ses bases Python (variables, boucles, fonctions, fichiers).

Évaluations en cours de formation

- QCM mi-parcours (20 questions) en fin de jour 2
- Évaluation sommative : QCM final de 30 questions en fin de jour 4
- Critère de réussite : obtenir 60 % au QCM final
- Attestation de fin de formation : remise à l'issue, mentionnant objectifs, durée et résultats

Satisfaction

- Questionnaire de satisfaction en fin de formation (formateur, contenu, logistique, atteinte des objectifs)

8. Accessibilité aux personnes en situation de handicap

Pythonia s'engage à rendre ses formations accessibles aux personnes en situation de handicap. Notre référent handicap est disponible pour étudier toute demande d'adaptation :

- Adaptation des supports pédagogiques (taille de police, contraste, format audio)
- Aménagement des conditions de passage (temps majoré, pauses adaptées)
- Orientation vers nos partenaires spécialisés : Agefiph, Cap Emploi, MDPH

Contact référent handicap : contact@pythonia.fr – 06 66 06 38 59

9. Engagement et suivi

- Suivi de l'assiduité : feuille d'émargement signée par demi-journée
- Prévention des abandons : suivi individualisé, adaptation du rythme, relance en cas d'absence
- Réclamations : toute réclamation peut être adressée par email à contact@pythonia.fr. Elle sera traitée sous 48h ouvrées.