

New

Analyse de données avec Python - Niveau Avancé

Visualisations interactives, statistiques avancées, gros volumes et automatisation

 Présentiel ou en classe à distance



3 jours (21 h)

Prix inter : 2.490,00 € HT
Forfait intra : 7.690,00 € HT

Réf.: BD557

La formation **Analyse de données avec Python - Niveau Avancé** permet de maîtriser les techniques avancées d'**analyse de données** et de **data visualisation** avec l'écosystème Python. Elle couvre les visualisations interactives, les **analyses statistiques avancées**, les **séries temporelles**, le traitement de **grands volumes de données** avec **Polars** et **Dask**, ainsi que l'industrialisation des analyses. Une formation idéale pour produire des traitements de données performants, reproductibles et adaptés aux environnements professionnels.

A qui s'adresse cette formation ?



Pour qui

- Data scientists, analystes expérimentés



Prérequis

- Connaissances en Python niveau intermédiaire
- **Disposez-vous des connaissances nécessaires pour suivre cette formation ? Testez-vous !**

Programme

1 - Visualisations avancées avec Matplotlib et Seaborn

- Anatomie avancée d'une figure Matplotlib : GridSpec, axes inset et twin axes pour combiner plusieurs échelles
- Personnalisation avancée des graphiques : annotations, texte mathématique LaTeX et transformations de coordonnées
- Création de styles réutilisables avec les stylesheets Matplotlib, les rcParams et les thèmes personnalisés
- Exploration multidimensionnelle avec Seaborn et FacetGrid : paramètres hue, col et row
- Création de visualisations 3D avec mpl_toolkits.mplot3d : nuages de points, surfaces et contours
- Conception d'animations avec FuncAnimation pour représenter l'évolution temporelle des données

Atelier
Construire une figure complexe avec GridSpec en combinant plusieurs sous-graphiques de tailles différentes

Créer un FacetGrid Seaborn croisant deux variables catégorielles avec les paramètres col et row afin d'analyser des tendances par sous-groupe

Ajouter des annotations et du texte mathématique LaTeX pour faciliter l'interprétation d'une régression

Créer une animation avec FuncAnimation pour représenter l'évolution d'une série temporelle

2 - Création de visualisations interactives avec Plotly

- Présentation de Plotly et de son approche déclarative : comparaison entre plotly.express et plotly.graph_objects
- Fonctionnalités interactives : zoom, survol, sélection, légende cliquable et sélection lasso
- Création de graphiques avancés : sunburst, treemap, parallel coordinates et diagramme de Sankey
- Conception de visualisations géographiques avec choropleth et scattermapbox

- Introduction à la création de dashboards composés de plusieurs vues coordonnées avec Dash
- Export et partage des visualisations : fichier HTML autonome et intégration dans une application web

Atelier

Créer un nuage de points interactif avec plotly.express

Configurer les fonctionnalités de survol et de sélection

Construire une visualisation géographique avec choropleth ou scattermapbox

Combiner plusieurs vues treemap, time série et heatmap dans une figure Plotly composée de plusieurs sous-graphiques

Exporter et partager le dashboard sous la forme d'un fichier HTML

3 - Régression et tests statistiques avancés avec Python

- Présentation de Statsmodels : modèles OLS et GLM, et formulation de type R avec la syntaxe $y \sim x_1 + x_2$
- Régression linéaire multiple : coefficients, R^2 , R^2 ajusté, p-values et intervalles de confiance
- Diagnostic d'un modèle de régression : analyse des résidus, hétéroscédasticité et multicollinéarité avec le VIF
- Régression logistique : variable binaire, odds ratio, courbe ROC et AUC
- Analyse de variance ANOVA à un ou plusieurs facteurs : statistique F et p-value
- Tests statistiques non paramétriques : Mann-Whitney, Wilcoxon et Kruskal-Wallis pour les distributions

Atelier

Estimer un modèle de régression linéaire multiple avec Statsmodels en utilisant une formulation de type R

Diagnostiquer le modèle à partir de l'analyse des résidus, d'un test de normalité et du VIF pour la multicollinéarité

Réaliser une ANOVA à deux facteurs afin de comparer les moyennes de plusieurs groupes

Appliquer un test de Mann-Whitney lorsque l'hypothèse de normalité des données n'est pas vérifiée

4 - Séries temporelles et analyses multivariées

- Décomposition d'une série temporelle : tendance, saisonnalité et résidu selon un modèle additif ou multiplicatif
- Étude de la stationnarité avec le test ADF, ou Augmented Dickey-Fuller
- Modélisation ARIMA : composantes AR, I et MA, et sélection des paramètres p, d et q
- Prévision avec un modèle ARIMA : forecast, intervalles de confiance et métriques d'évaluation
- Analyse en composantes principales (PCA) : réduction de dimension et variance expliquée
- Visualisation des composantes principales et interprétation de la contribution des variables

Atelier

Décomposer une série temporelle de ventes en tendance, saisonnalité et résidu avec seasonal_decompose

Tester la stationnarité avec le test ADF et différencier la série lorsque cela est nécessaire

Estimer un modèle ARIMA, générer une prévision à 30 jours et représenter les intervalles de confiance

Réaliser une PCA sur un jeu de données multidimensionnel

Visualiser la variance expliquée et les composantes principales

5 - Manipulation de grands volumes de données avec Polars et Dask

- Limites de Pandas pour le traitement de grands volumes : exécution sur un seul thread et contraintes liées à la mémoire
- Présentation de Polars : moteur développé en Rust, API expressive et performances annoncées de 5 à 10 fois supérieures à Pandas
- Fonctionnement de Polars : modes lazy et eager, expressions, agrégations groupby et jointures optimisées
- Présentation de Dask : parallélisation des traitements et exécution out-of-core sur des jeux de données dépassant la mémoire vive disponible
- Utilisation de Dask DataFrame : API proche de Pandas, partitionnement des données et exécution différée
- Présentation de PySpark pour les volumes de données de plusieurs téraoctets et les environnements distribués

Atelier

Charger un fichier CSV de plusieurs gigaoctets avec Polars en mode lazy et comparer les temps de traitement avec Pandas

Filtrer, agréger et joindre les données avec l'API Polars, ses expressions et ses opérations groupby

Convertir un pipeline Pandas existant en pipeline Polars équivalent et mesurer le gain obtenu

Traiter avec Dask DataFrame un jeu de données dont le volume dépasse la mémoire vive disponible

6 - Automatisation et industrialisation des analyses Python

- Transformation d'un notebook d'analyse exploratoire en script Python réutilisable

- Création de scripts en ligne de commande avec argparse ou Click : paramètres, options et aide intégrée
 - Exécution paramétrée de notebooks avec Papermill pour automatiser le reporting
 - Mise en place de la journalisation et de la gestion des erreurs avec le module logging, les niveaux de logs et les fichiers structurés
 - Planification de scripts simples avec cron et présentation des orchestrateurs Airflow et Prefect
 - Packaging et bonnes pratiques : environnement virtuel, fichier requirements.txt et structuration d'un projet Python
- Atelier

Transformer un notebook d'analyse en script Python paramétrable en ligne de commande avec argparse

Exécuter avec Papermill un notebook de reporting paramétré pour trois segments différents

Ajouter au script un système de journalisation et de gestion des erreurs avec le module logging

Planifier l'exécution quotidienne du script avec cron

Contrôler les fichiers de logs produits par le pipeline automatisé



Les objectifs de la formation

- Concevoir des visualisations de données avancées pour explorer et communiquer les résultats
- Mettre en oeuvre des techniques d'analyse de données adaptées à des problématiques métier
- Automatiser les traitements et les analyses de données à l'aide de scripts Python
- Produire des analyses reproductibles et exploitables pour la prise de décision



Evaluation

- Pendant la formation, le formateur évalue la progression pédagogique des participants via des QCM, des mises en situation et des travaux pratiques. Les participants passent un test de positionnement avant et après la formation pour valider leurs compétences acquises.



Les points forts de la formation

- Une approche pédagogique fondée sur des cas d'usage concrets, permettant de maîtriser les techniques avancées de data science avec Python, de l'analyse statistique à l'automatisation des traitements
- Une formation couvrant les principales méthodes d'analyse avancée des données : régression, ANOVA, séries temporelles ARIMA, analyse en composantes principales (ACP/PCA), visualisation interactive et optimisation des performances
- Des ateliers pratiques mettant en oeuvre les outils de référence de l'écosystème Python (statsmodels, Plotly, Polars, Dask et Papermill) pour analyser, visualiser et industrialiser des traitements sur de grands volumes de données



Dates et villes 2026 - Référence BD557



Dernières places disponibles



Session garantie

A distance

du 16 nov. au 18 nov.

du 21 déc. au 23 déc.

Paris

du 16 nov. au 18 nov.

du 21 déc. au 23 déc.