



# — Phirio —

## Visualisation avancée de données avec Python

LY005

Durée: 2 jours

2 030 €

3 au 5 février  
5 au 7 mai

27 au 29 août  
26 au 28 novembre

### Public :

Développeurs souhaitant mettre en valeur les données scientifiques avec un support visuel, Data analysts, Data scientists.

### Objectifs :

Savoir traiter les données scientifiques brutes pour leur visualisation  
Savoir utiliser les bibliothèques graphiques de python pour visualiser des données  
Graphiques spécifiques : graphes dynamiques, cartographie

### Connaissances préalables nécessaires :

Avoir suivi la formation "Python Initiation" ou avoir un niveau équivalent et avoir une pratique régulière du langage Python

### Programme :

#### Visualisation de données

Contexte de la visualisation de données scientifiques et pièges à éviter  
Concepts essentiels de la communication visuelle (couleurs, taille, forme, type de graphique, ...)  
Architecture de présentation des données (DPA)  
La valorisation des données. Les indicateurs de performance clés (KPI).  
Objectifs et règles de conception : clarté, pertinence, cohérence, stimulation.  
Principes de perception des informations.  
Types de visualisation de base : graphiques linéaires, diagrammes en secteurs, cartes de zones, frise chronologique, diagrammes de dispersion, arbres, pyramides des populations  
Types évolués : infographie, nuages de bulles, graphiques à puces, cartes de chaleur, graphiques de séries chronologiques

Atelier : mise en évidence de la pertinence du type de graphiques :  
comparaison camemberts, barres, chandeliers japonais, ...

#### Personnalisation des graphiques avec Matplotlib

Fonctionnement de matplotlib : les backends, graphiques interactifs, polices, gestion des événements, performances.  
Exploration du package pour créer des graphiques sur différents types de données (qualitatives, quantitatives, séries temporelles, 3 dimensions)  
Affiner et compléter les graphiques (échelle, valeurs aberrantes, barres d'erreur, etc.)  
Les types : bar, scatter, plot, boxplot, fill\_between, imshow, tricontour, quiver  
Personnalisation de graphiques (légende, points remarquables avec flèche et texte en LaTeX dans le graphique, modification du style de graphique)

Atelier : création de graphiques personnalisés. Utilisation des styles Matplotlib



# — Phirio —

Atelier : détection d'anomalie par l'utilisation d'animations

## Packages spécialisés

Transformer des données avec Pandas (calculs d'agrégats, traitement des valeurs manquantes ou incohérentes, gestion des dates, etc.)

Atelier : nettoyage, préparation et regroupement de données de températures. Mise en œuvre des bfill, ffill, ... Gestion des NaN.

Gestion des données temporelles. Préparation des données pour visualisation. Le package Seaborn pour les données statistiques (box plot, pair plot, violin plot, matrices de graphiques, cartes thermiques, etc.)

Atelier : installation du paquet et création d'une carte thermique sur les données précédentes.

Cartographie (package cartopy et/ou Folium)

Atelier : ajout d'une planisphère sur la carte thermique précédente. Visualisation des points chauds de la planète en fonction des années.

## Interactivité et gros volumes de données

Création de dashboards simples (graphiques et boutons simples permettant l'action de l'utilisateur)  
Dashboards interactifs et partageables (par exemple, lien avec un notebook IPython ou Jupyter)  
Création de graphiques web interactifs avec les packages Bokeh et Plotly  
Applications web interactives avec Dash (framework basé sur Plotly)  
Les APIs Bokeh et Plotly

Atelier : mise en œuvre de Bokeh et Plotly. Création d'applications web avec Dash. Développement d'un graphique interactif reliant les isothermes de la carte précédente

Visualisation dans un navigateur.  
Passage à l'échelle, présentation de l'écosystème HoloViz : datashader, geoviews, panel.  
Gros volume de données avec datashader ou HoloViz.

Atelier : intégration de données vents, visibilité au graphique précédent.

Mise en évidence de la tenue à la charge.