

Chapitre 3

Outils et technologies utilisés en Data Science

Pré[paré-senté] par :
Dr. Bilal Dendani



جامعة باجي مختار - عنابة
BADJI MOKHTAR - ANNABA UNIVERSITY

Dr. DENDANI Bilal



Chapitre 3 : Outils et technologies utilisés en Data Science

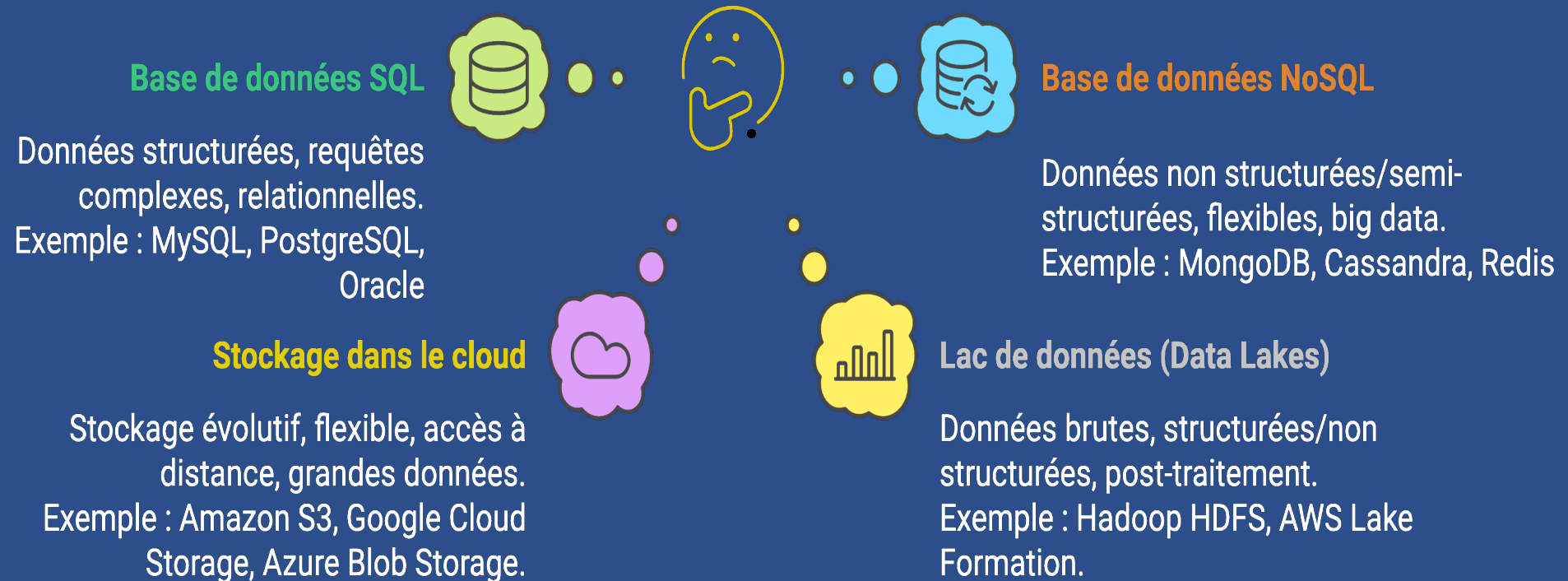
- Les outils de stockage de données
- Les outils de préparation de données
- Les outils de visualisation de données
- Les outils IDE notebooks
- Les plateformes complètes de Data science

Introduction

- En science de données, la maîtrise des **outils** et **technologies** est essentielle pour **transformer** les données brutes en informations **exploitables**.
- Les Data Scientists utilisent un ensemble varié d'outils pour chaque étape du cycle de vie de sciences de données, depuis le stockage jusqu'à l'analyse et la visualisation.
- Ces outils facilitent la manipulation de grandes quantités de données, automatisent les tâches de préparation, et permettent des analyses avancées.

Les outils de stockage de données

Quelle solution de stockage de données choisir ?



Bases de données relationnelles (SQL)

- Utilisent un modèle structuré basé sur des tables et des relations entre elles. Elles sont largement utilisées pour **stocker** et **interroger** des **données bien organisées**.
- Organisent les données en tables avec des colonnes et des lignes, chaque table représentant une entité (par exemple, Clients, Ventas).
- Le langage de requête utilisé est le Structured Query Language (SQL) pour manipuler et interroger les données.
- **MySQL, PostgreSQL, Oracle.**



Bases de Données **NoSQL** (Not Only SQL)

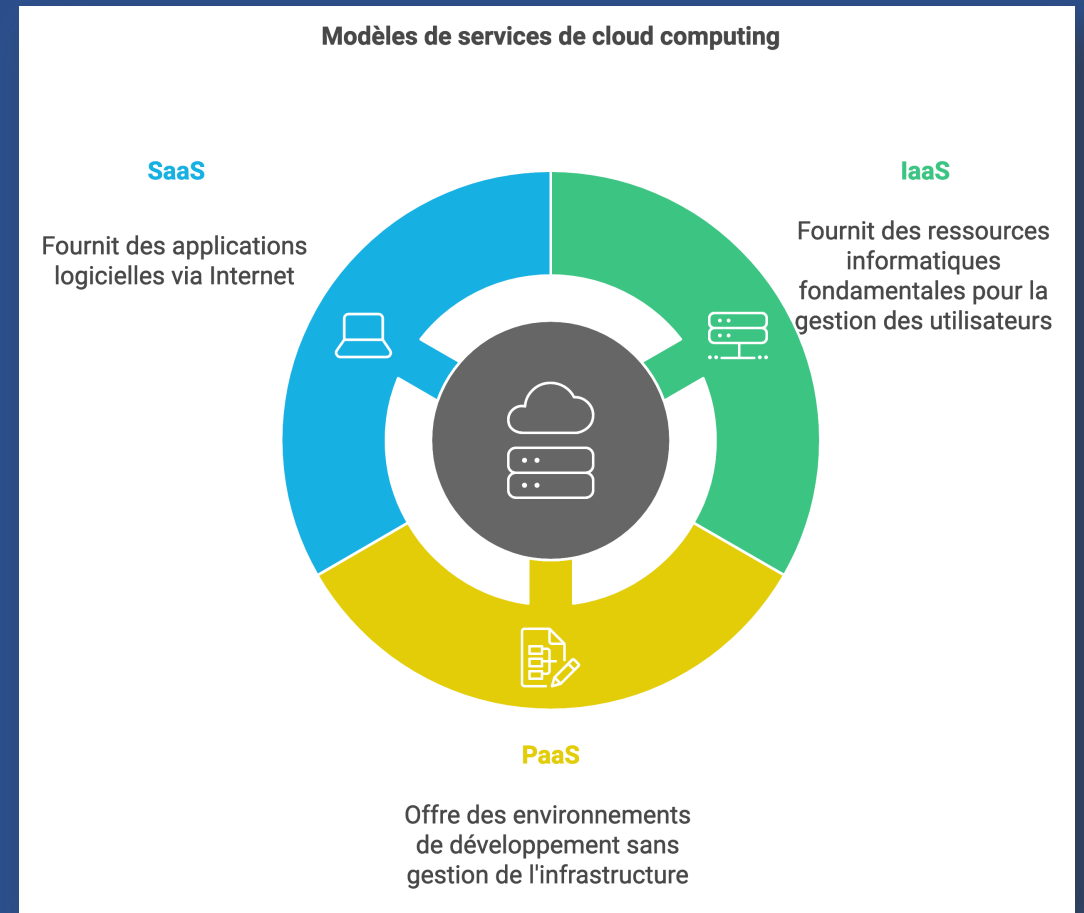
- Offrent une grande flexibilité, permettant de stocker des données non structurées ou semi-structurées, comme les documents JSON, les publications sur les réseaux sociaux ou les données de capteurs de manière plus scalable.
- NoSQL est utilisé pour les données à grande échelle, caractérisé par sa vitesse et sa flexibilité élevées.
- Elles sont conçues pour la scalabilité et la flexibilité, ce qui les rend adaptées aux applications de big data et aux analyses en temps réel.
- **Exemples : MongoDB, Redis, Cassandra.**



<https://www.datacamp.com/blog/nosql-databases-what-every-data-scientist-needs-to-know>

Stockage dans le cloud

- Le stockage dans le cloud permet un accès distant, une haute disponibilité et une scalabilité selon les besoins.
- Exemples: Amazon S3, Google Cloud Storage, Azure Blob Storage.



Lacs de Données (Data Lakes)

- Les lacs de données sont utilisés pour le stockage de données brutes structurés ou non-structurés.
- Utilisation pour les analyses avancées et le machine learning.
- Technologies associées : HDFS, AWS Lake Formation, Azure Data Lake.

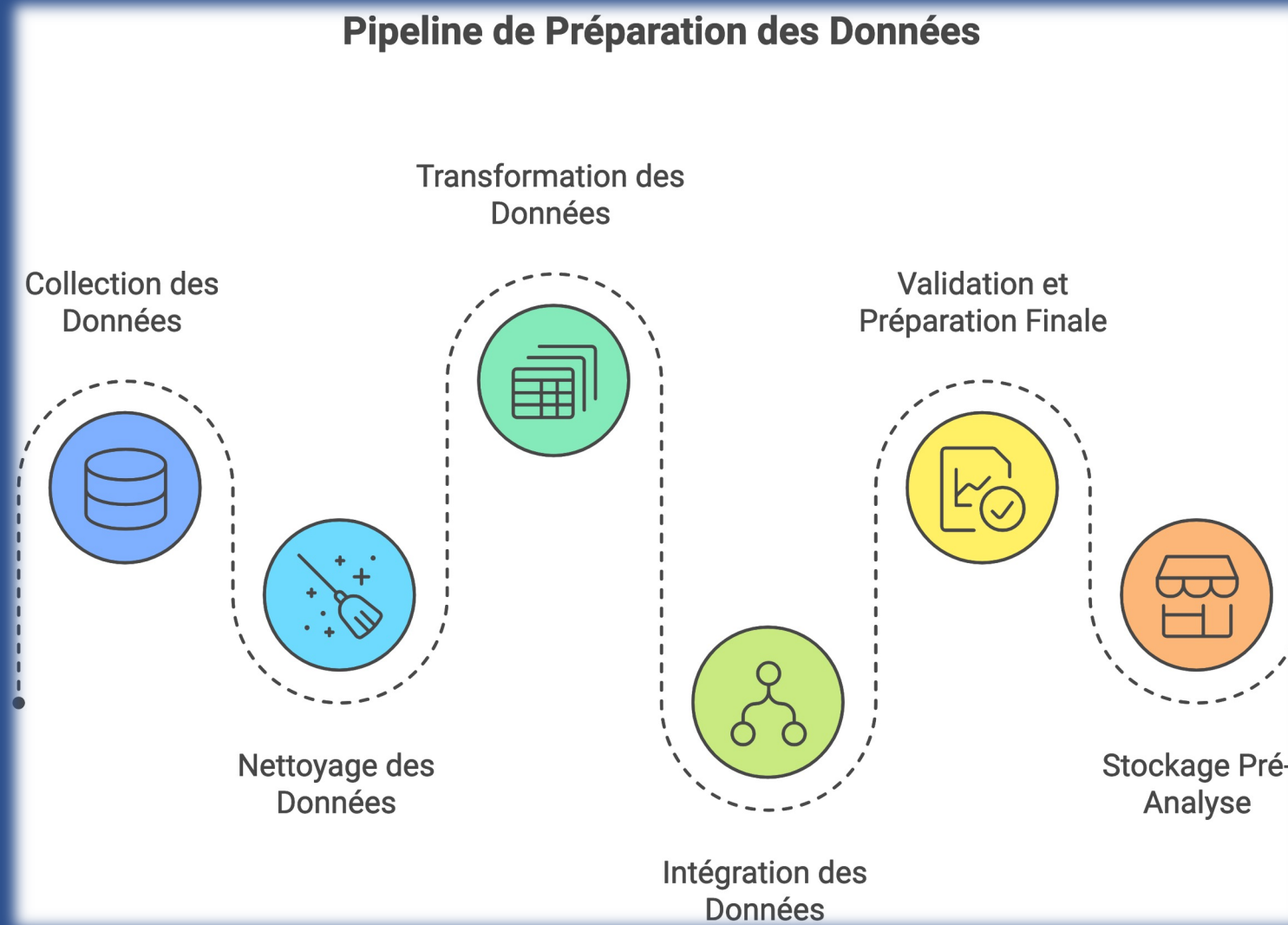


Outils de Préparation de Données en Science des Données

- La préparation des données est une étape clé du processus de science des données.
- Elle vise à nettoyer, transformer, et intégrer les données pour assurer leur qualité et leur utilité.
- Dans ce partie, nous explorerons différents outils utilisés pour chacune de ces étapes



Pipeline de préparation de données



Outils de collect de données

Outils de Collecte de Données

Stockage de Données

Bibliothèques pour lire et traiter des données à partir de divers formats de fichiers.

- Bibliothèque Pandas : lire fichier CSV
- Spark : Charger de données de grandes quantités

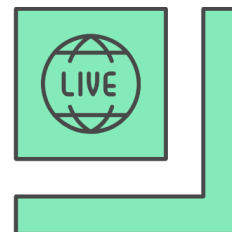


Bases de Données SQL

Outils pour extraire des données des bases de données relationnelles en utilisant des requêtes structurées.

Outils ETL(Extract, Transform, Load)

- Logiciels pour automatiser la collecte, la transformation et l'intégration des données.
- Outils :Talend, Apache Nifi

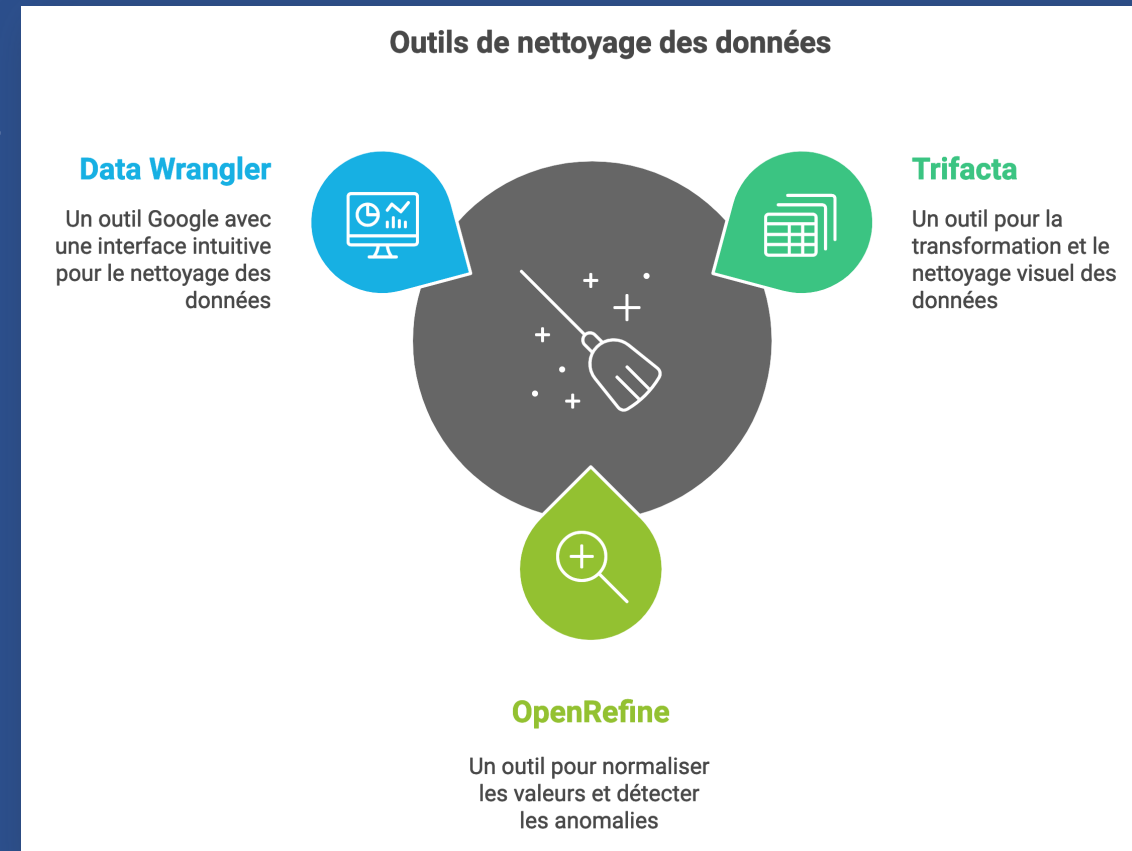


APIs

- Bibliothèques pour la collecte de données en temps réel via des requêtes HTTP.
- Bibliothèque Python Request pour collecter des données en temps réels

Outils de Nettoyage des Données

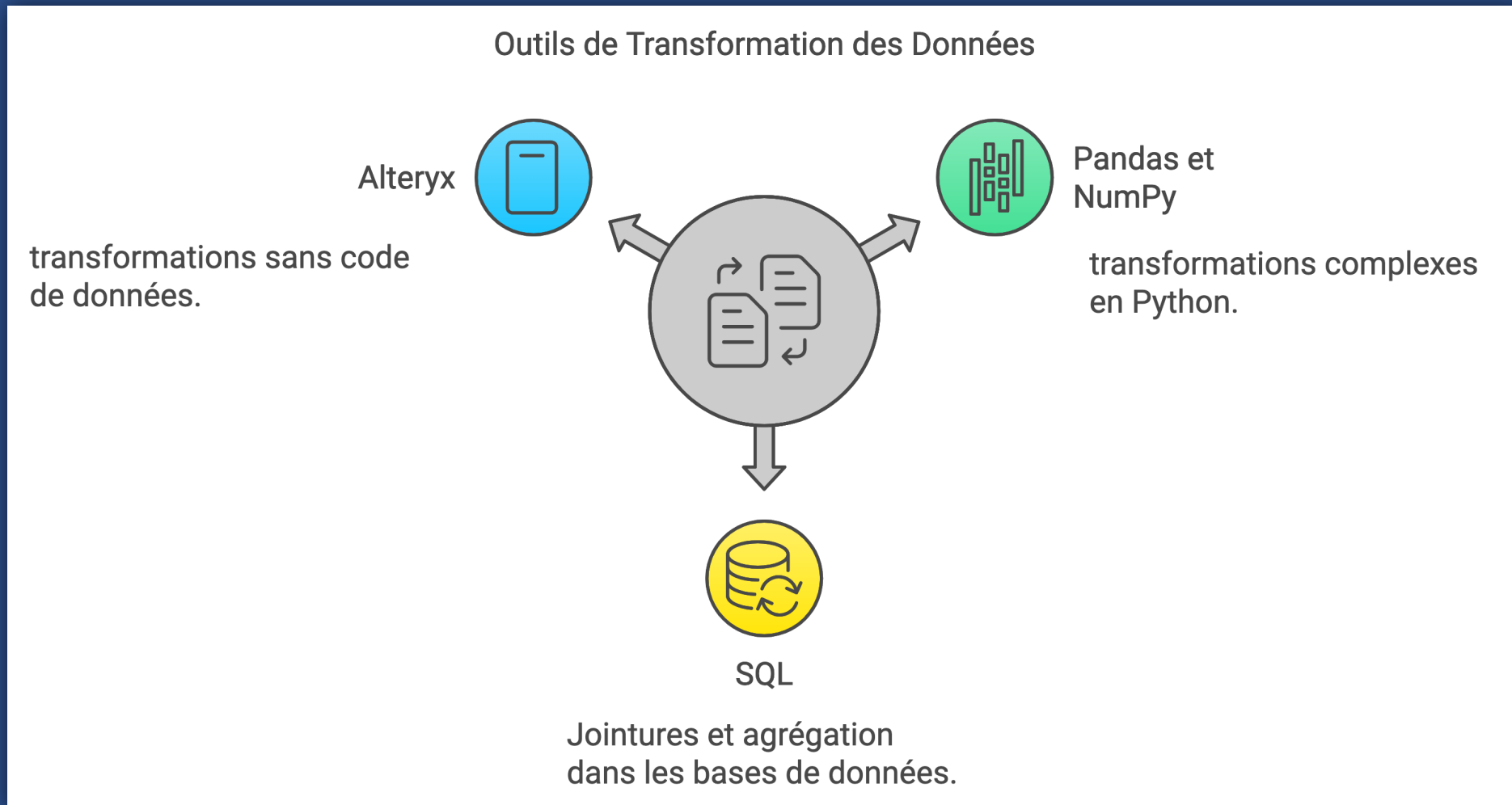
- Le **nettoyage de données** est une étape essentielle dans le processus de science des données.
- Le nettoyage de données permet d'améliorer la Qualité des Données.
- Le nettoyage permet d'éliminer ou de corriger les valeurs aberrantes ou erronées qui peuvent **biais**er les résultats des modèles de machine learning et des analyses statistiques.



Outils de Transformation des Données

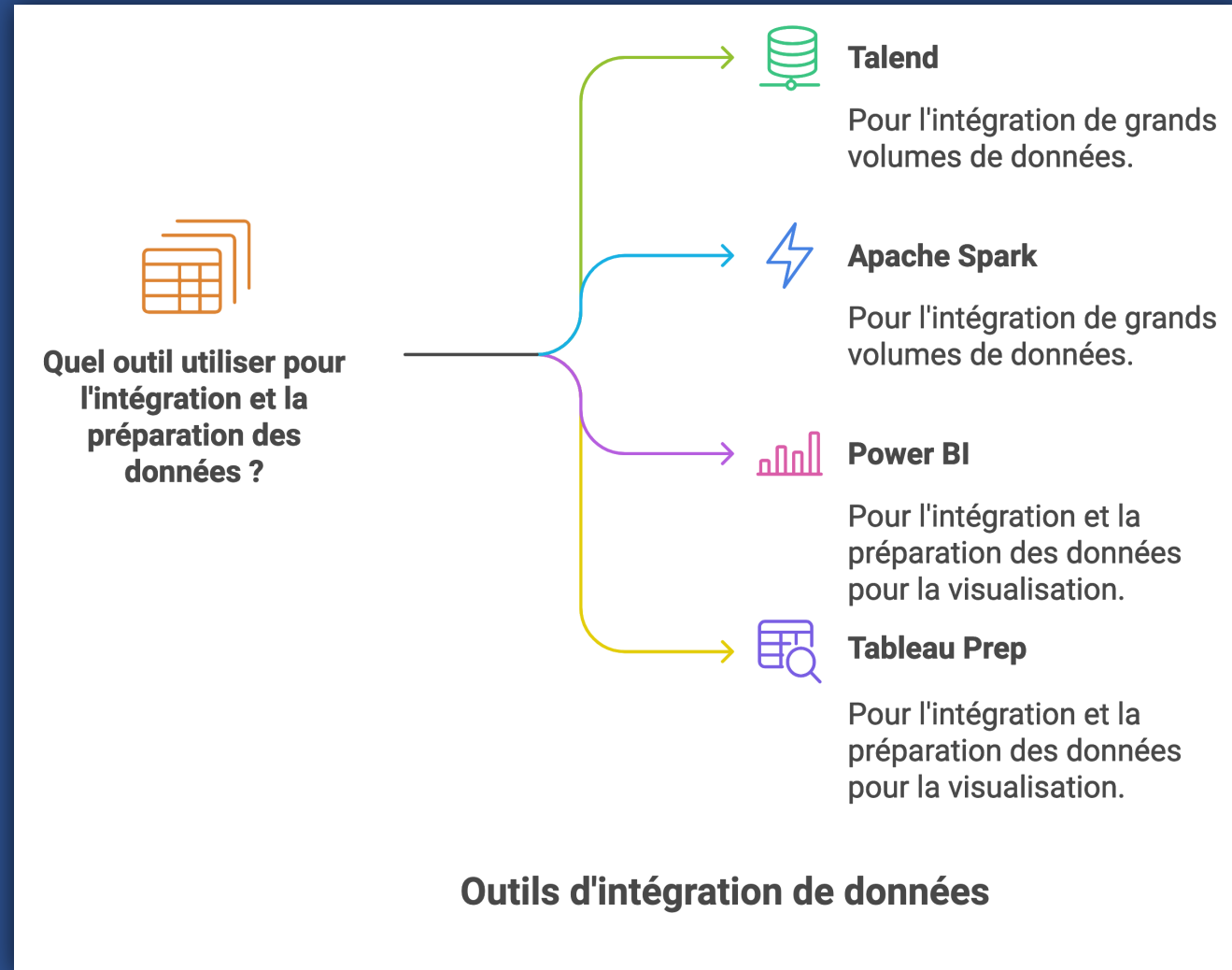


Outils de transformation de données



Outils d'intégration de données

- L'intégration de données est le processus de **combiner** des données provenant de **différentes sources** pour créer une vue unifiée et cohérente.
- Cela implique la collecte, la transformation et la consolidation des données afin qu'elles puissent être utilisées efficacement pour l'analyse et la prise de décision.



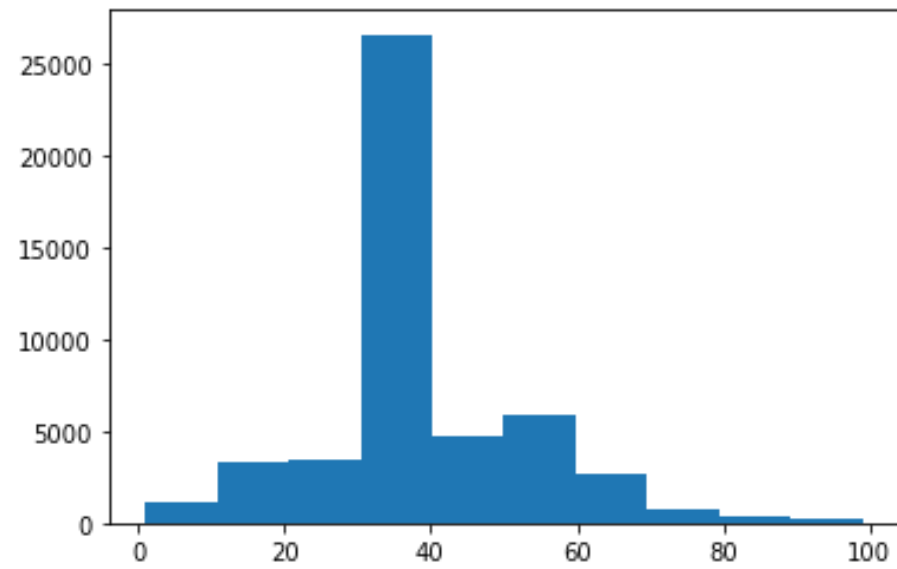
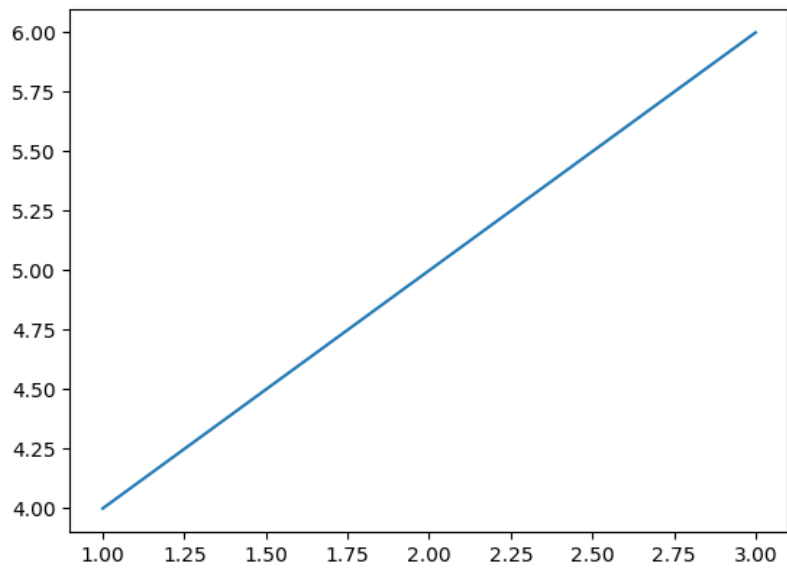
Outils de visualisation de données

- En data science, les outils de visualisation de données jouent un rôle crucial pour explorer les données, identifier les tendances et communiquer les résultats. Voici les principaux outils utilisés, classés en fonction de leur type et de leurs usages :
- **1. Bibliothèques de visualisation en Python :**

Python est l'un des langages les plus populaires en data science, et il offre une riche collection de bibliothèques pour la visualisation.

Matplotlib

- Une bibliothèque fondamentale pour créer des graphiques statiques, animés ou interactifs.
- Utilisées pour construire des courbes, histogrammes, graphiques à barres, nuages de points.
- Leur points forts est sa haute personnalisation, adaptable à d'autres bibliothèques comme pandas et NumPy.



Boxplot

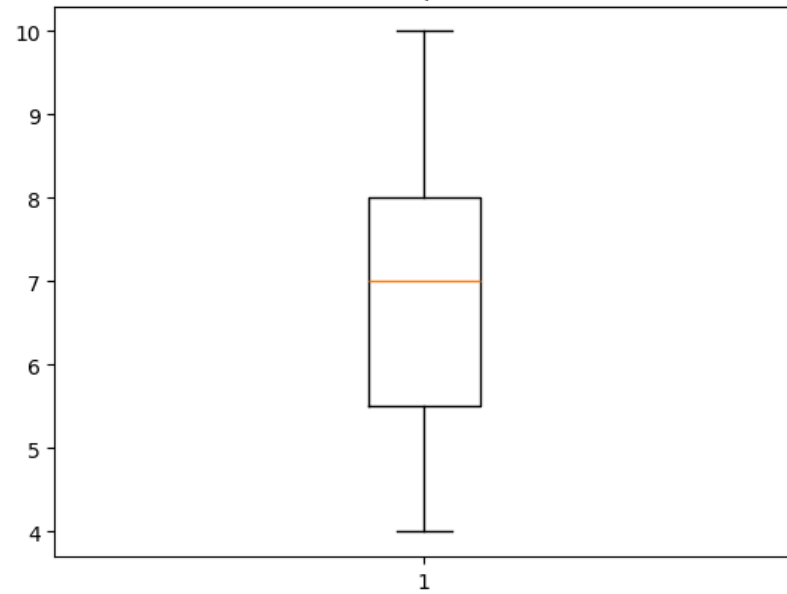
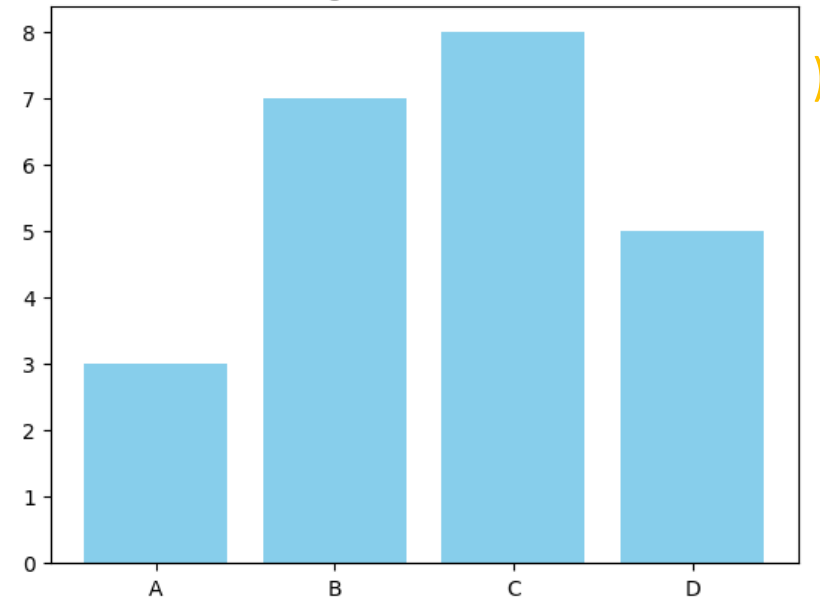


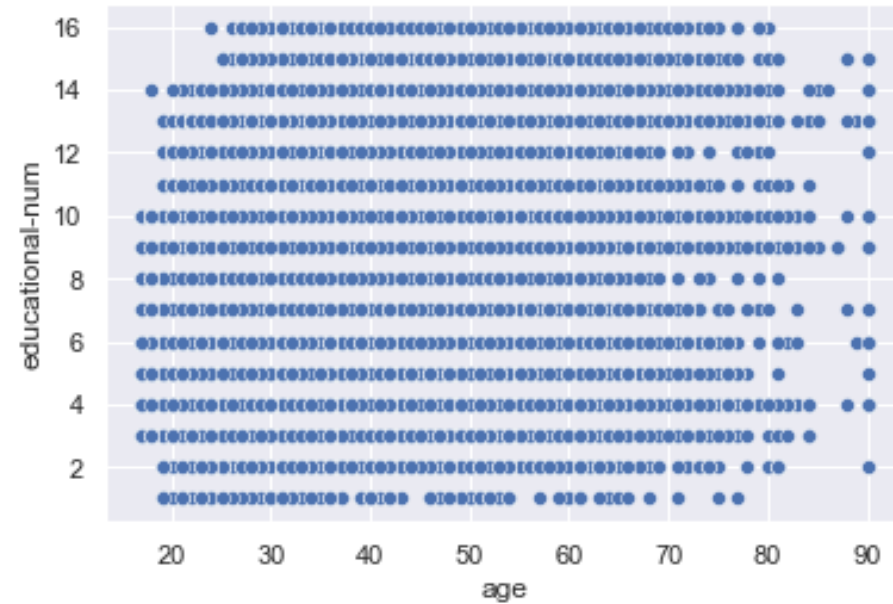
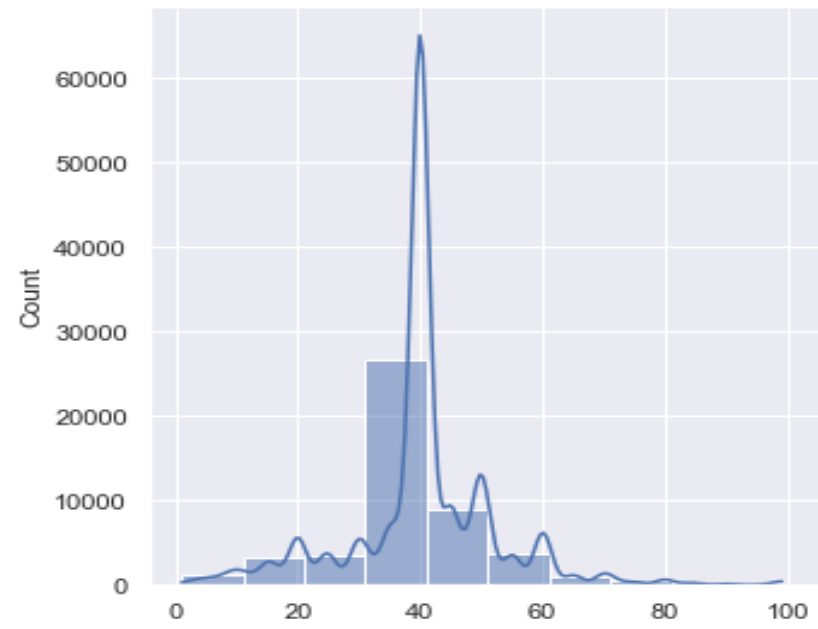
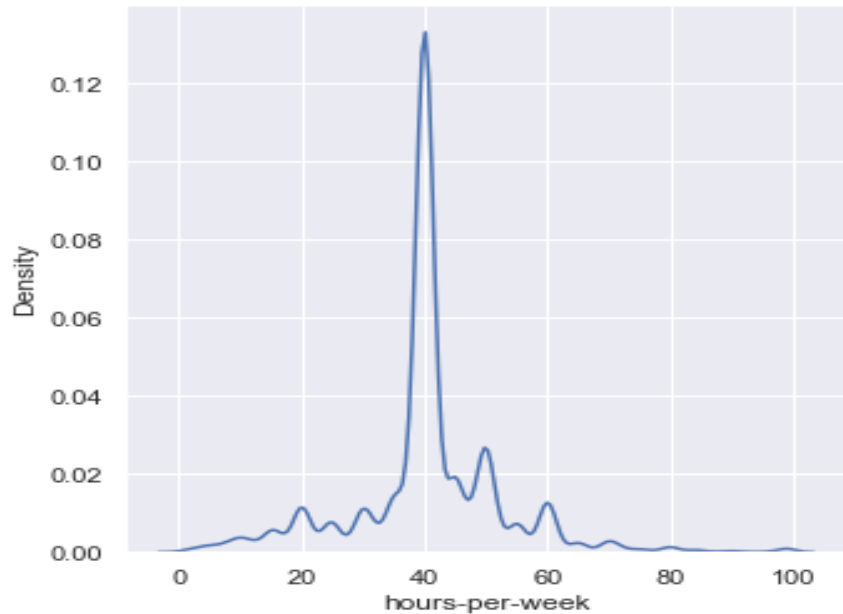
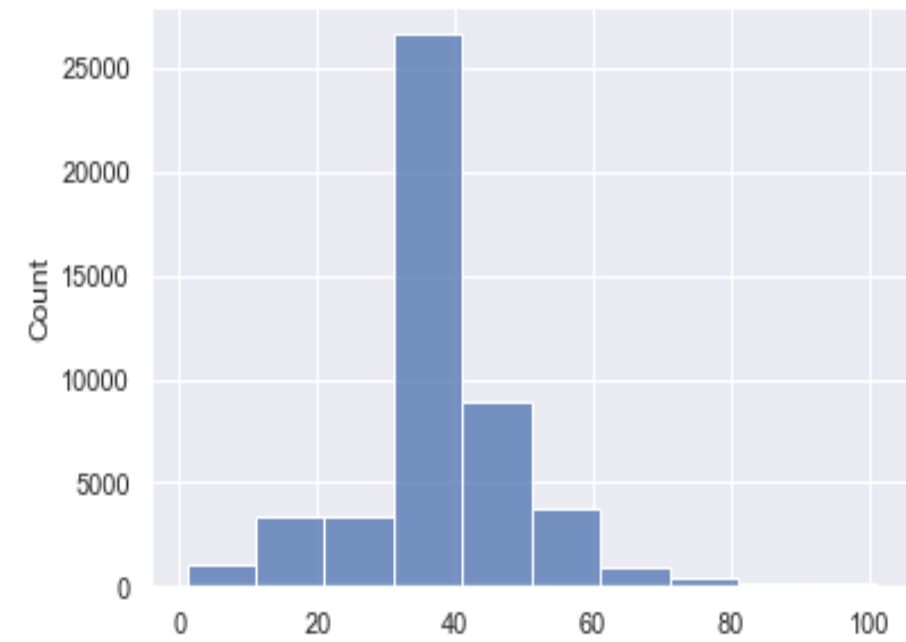
Diagramme en Barres



<https://colab.research.google.com/drive/1n4uImR1BCV8ZJI1AJCHJvQMMcMANWuy5>

Seaborn

- Basé sur Matplotlib, Seaborn est conçu pour des graphiques statistiques avancés.
- Usages Visualisation de distributions, heatmaps, régressions.
- Points forts : Esthétiques par défaut, simplification des tâches complexes.



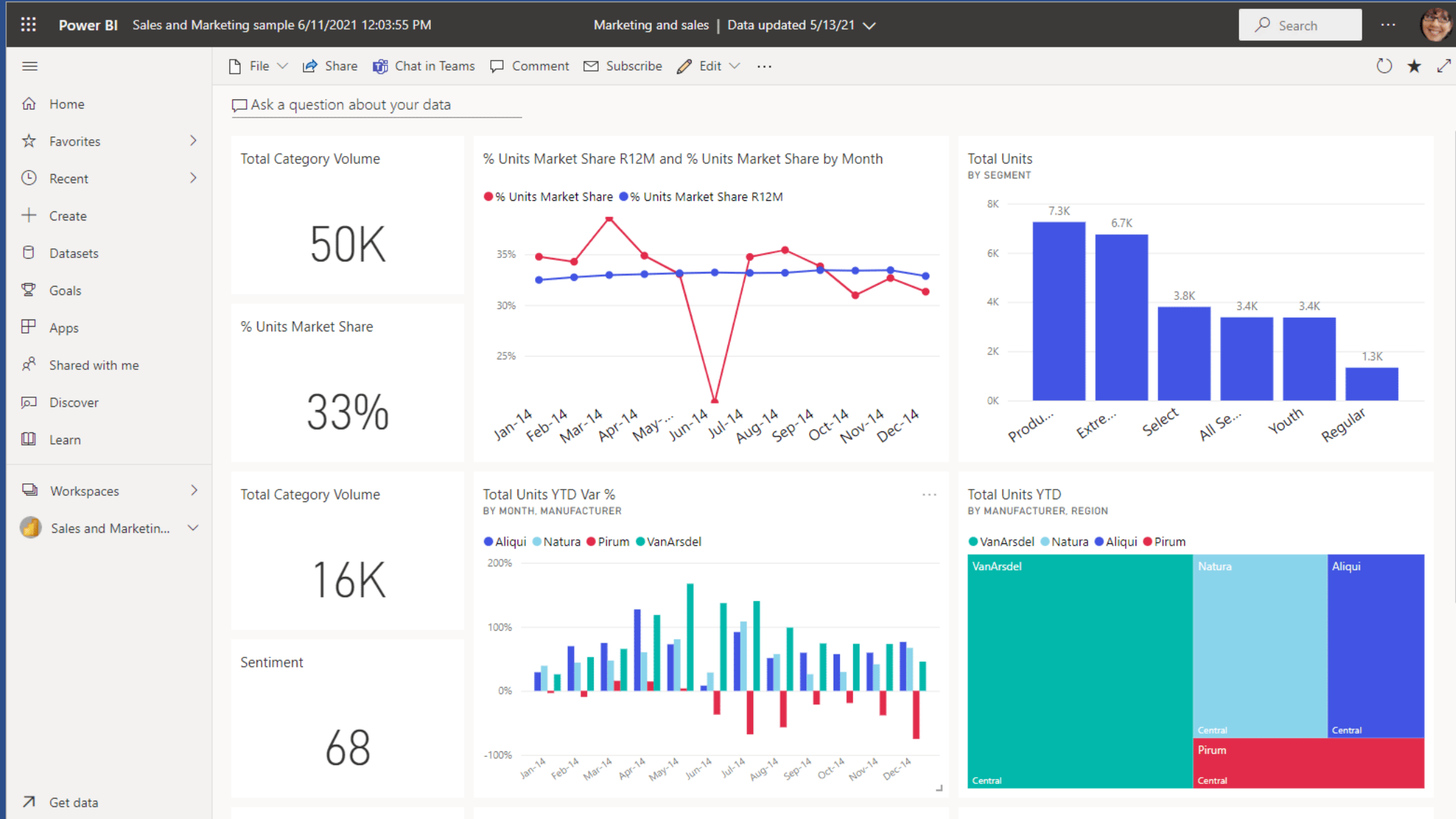
Plotly

- Outil interactif pour créer des visualisations dynamiques et des tableaux de bord.
- Utilisés pour créer des cartes interactives, des graphiques 3D, des diagrammes complexes.
- Les points forts de cette bibliothèque réside dans sa compatibilité avec Dash pour créer des tableaux de bord.

Logiciels de visualisation de données : Tableau



Logiciels de visualisation de données : PowerBI



Google Data Studio

Google News Initiative Resources About ? Q English English My Account

< DATA JOURNALISM


Lesson 11 of 13 Data Studio: Make interactive data visualizations

Data Studio: Make interactive data visualizations


Give life to your datasets by creating powerful interactive visualizations with an easy-to-use studio.

[Download Lesson](#)


Geo map




Line



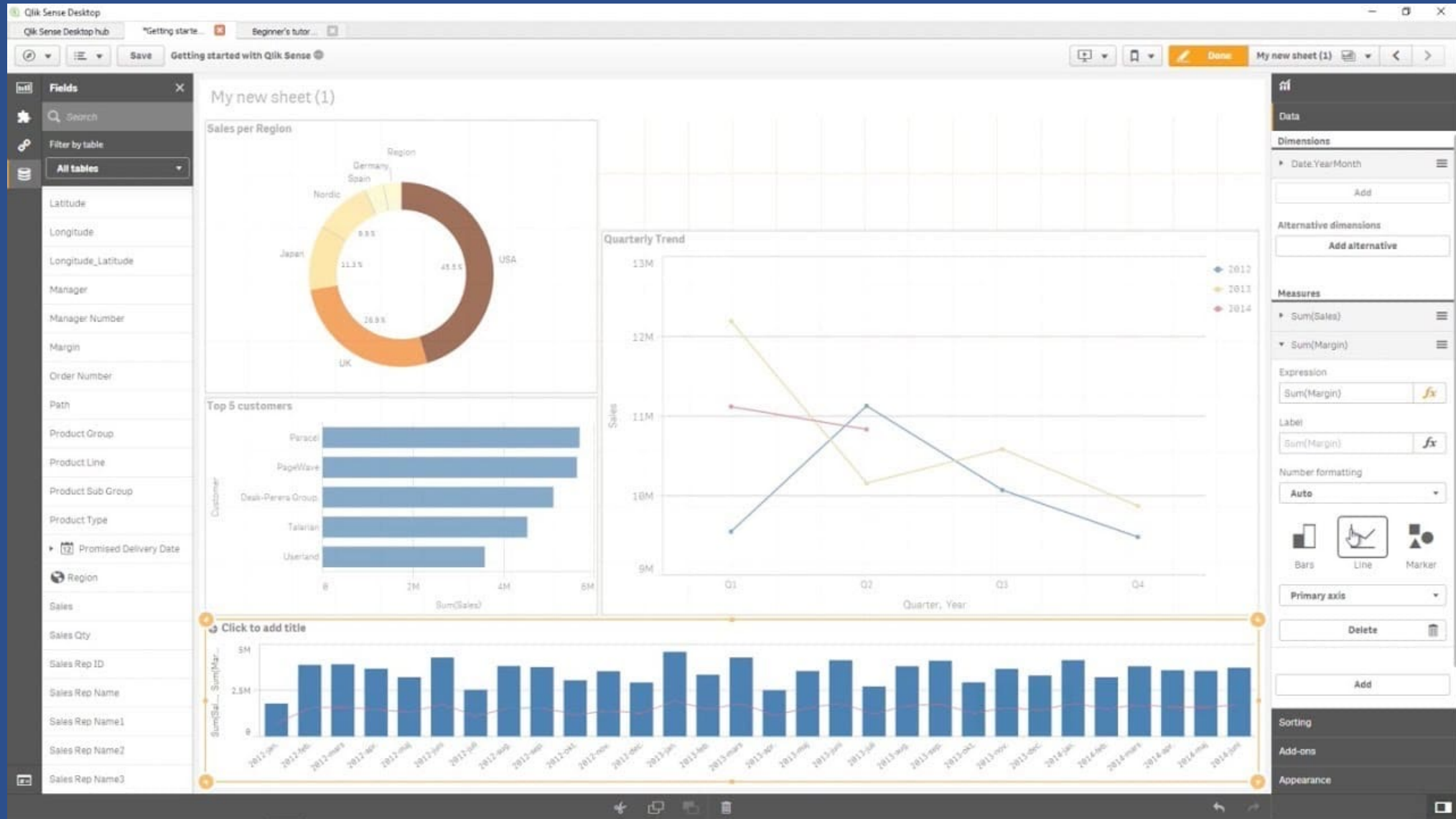
Area



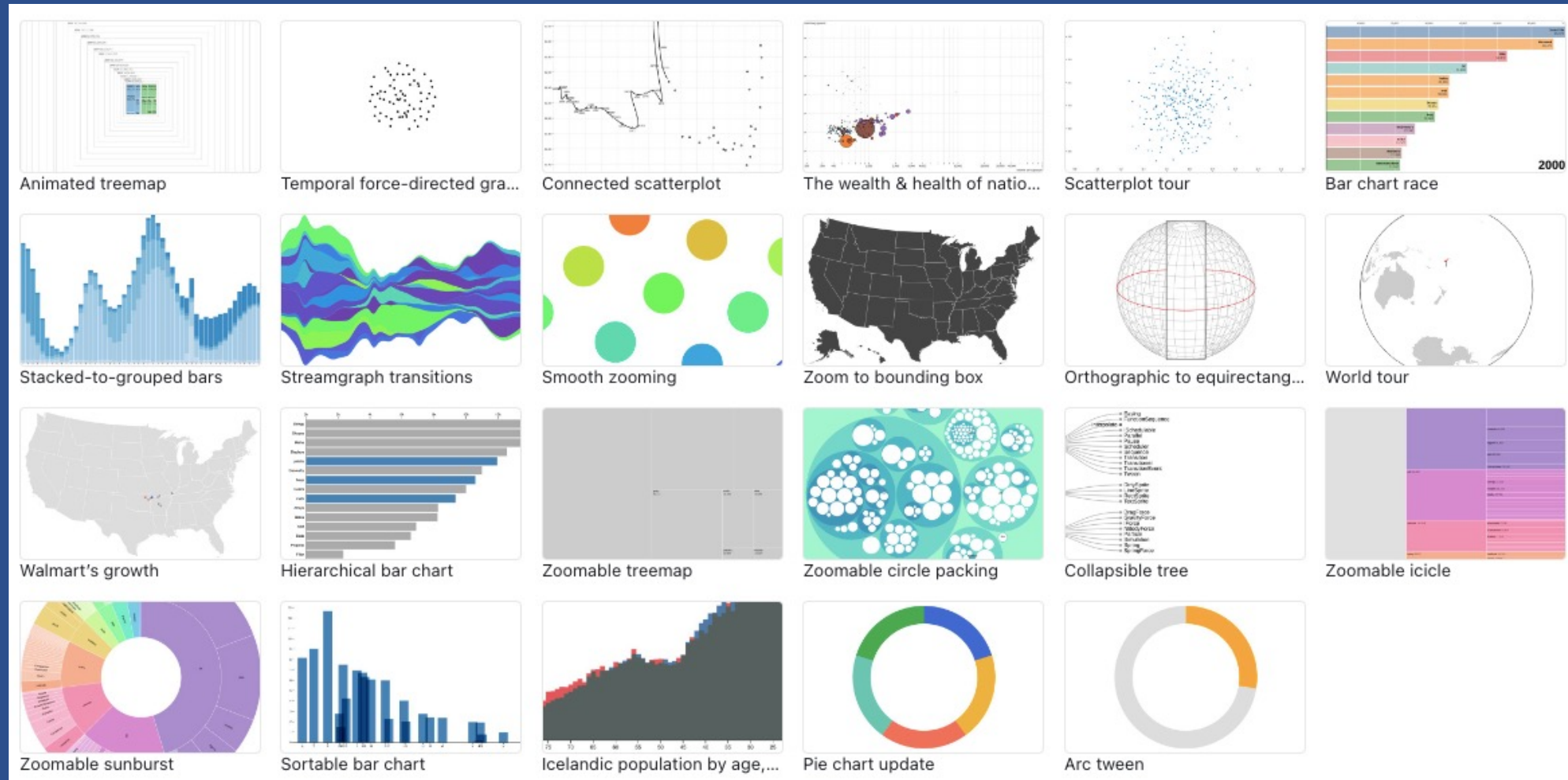
Scatter



Qlik Sense



D3.js



Les outils de développement IDE Notebook

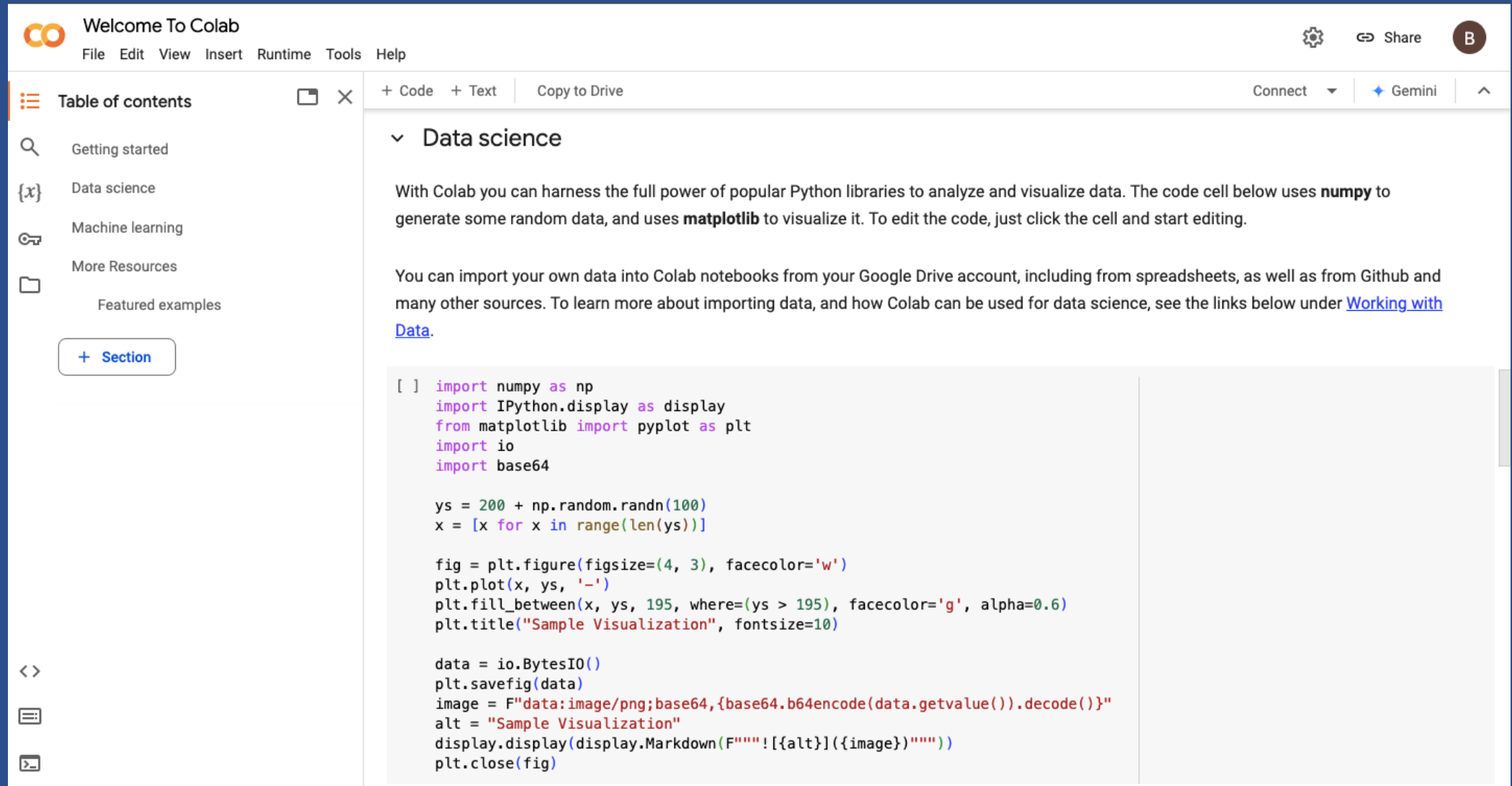
- Les outils IDE notebooks (Integrated Development Environment Notebooks) sont des environnements interactif.
- Permettant d'écrire et d'exécuter du code tout en intégrant des visualisations, des explications textuelles et des analyses de données.
- Ils sont particulièrement utiles en science des données pour explorer des datasets, tester des algorithmes, et présenter des résultats.

Les outils IDE notebooks: Jupyter notebook et Jupyter Lab

The image displays the Jupyter Lab interface, which is a web-based IDE for data science. It features a multi-view environment where users can work with code, text, and data simultaneously.

- File Browser:** Located on the left, it shows a directory structure with files like 'Linear Regression.ipynb' and 'Lorenz.ipynb'.
- Notebook Editor:** The main area shows a notebook titled 'In Depth: Linear Regression'. It contains text explaining linear regression models and a code cell with a plot of data points.
- Launcher:** A central panel provides quick access to different environments: Python 3, C++11, C++14, C++17, Julia 1.0, phylogenetics (Python 3.7), and R.
- Output View:** On the right, it displays the output of a code cell, showing a scatter plot of 'Seattle Weather: 2012-2015' with 'Maximum Daily Temperature (C)' on the y-axis and 'Date' on the x-axis. Below the plot is a horizontal bar chart showing the 'Number of Records' for each month.
- Open Notebooks:** Several other notebooks are open in the background, including 'Julia.ipynb' (showing a plot of 'x' vs 'y'), 'Lorenz.ipynb' (showing the Lorenz attractor equations), and 'R.ipynb' (showing a scatter plot of 'Sepal.Length' vs 'Petal.Length').

Google Colab



The screenshot shows the Google Colab interface. At the top, there is a "Welcome To Colab" header with a menu (File, Edit, View, Insert, Runtime, Tools, Help) and a user profile icon. On the left, a "Table of contents" sidebar lists sections like "Getting started", "Data science", "Machine learning", and "More Resources". The main area is titled "Data science" and contains two paragraphs of text. The first paragraph explains that Colab allows using Python libraries like **numpy** and **matplotlib** for data analysis and visualization. The second paragraph discusses importing data from Google Drive, Github, and other sources, with a link to "Working with Data". Below the text is a code cell containing Python code that generates random data and creates a plot with a green shaded area for values above 195. The code uses `numpy`, `IPython.display`, `matplotlib.pyplot`, and `io` modules.

```
[ ] import numpy as np
import IPython.display as display
from matplotlib import pyplot as plt
import io
import base64

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]

fig = plt.figure(figsize=(4, 3), facecolor='w')
plt.plot(x, ys, '-')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)
plt.title("Sample Visualization", fontsize=10)

data = io.BytesIO()
plt.savefig(data)
image = F"data:image/png;base64,{base64.b64encode(data.getvalue()).decode()}"
alt = "Sample Visualization"
display.display(display.Markdown(F""""!{{alt}}({image})"""))
plt.close(fig)
```

Spyder

The screenshot displays the Spyder IDE interface with the following components:

- Left Panel (Project Explorer):** Shows a tree view of the project structure. The 'Plots' folder is expanded, showing methods like `get_name`, `get_description`, `get_icon`, `register`, `unregister`, `switch_to_plugin`, `current_widget`, `add_shellwidget`, `remove_shellwidget`, and `set_shellwidget`. Other folders include `plot_example.py` and `plugin.py`.
- Center Panel (Code Editor):** Displays the source code for `plugin.py`. The code includes a docstring, imports, and a `Plots` class that inherits from `SpyderDockablePlugin`. The class defines methods for `get_name`, `get_description`, `get_icon`, and `register`.
- Right Panel (Variable Explorer):** Shows a table of variables in the current environment. The table has columns for Name, Type, Size, and Value.
- Bottom Panel (Plots):** Contains two plots: a 3D surface plot on the left and a polar plot on the right.

Name	Type	Size	Value
a	foo	1	foo object of __main__ module
filename	str	53	/Users/Documents/spyder/spyder/tests/test_dont_use.py
i	bool	1	True
my_set	set	3	{1, 2, 3}
r	float	1	6.46567886443
t	tuple	5	('abcd', 745, 2.23, 'efgh', 78.2)
thisdict	dict	3	{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
tinylist	list	2	[123, 'efgh']
x	Array of int64 (2,)		[1 2]
y	timedelta	1	2 days, 0:00:00

```
1  | # -*- coding: utf-8 -*-
2  | #
3  | # Copyright © Spyder Project Contributors
4  | # Licensed under the terms of the MIT License
5  | # (see spyder/__init__.py for details)
6  |
7  | """
8  | Plots Plugin.
9  | """
10 |
11 | # Third party imports
12 | from qtpy.QtCore import Signal
13 |
14 | # Local imports
15 | from spyder.api.plugins import Plugins, SpyderDockablePlugin
16 | from spyder.api.translations import get_translation
17 | from spyder.plugins.plots.widgets.main_widget import PlotsWidget
18 |
19 | # Localization
20 | _ = get_translation('spyder')
21 |
22 |
23 |
24 | class Plots(SpyderDockablePlugin):
25 |     """
26 |     Plots plugin.
27 |     """
28 |     NAME = 'plots'
29 |     REQUIRES = [Plugins.IPythonConsole]
30 |     TABIFY = [Plugins.VariableExplorer, Plugins.Help]
31 |     WIDGET_CLASS = PlotsWidget
32 |     CONF_SECTION = NAME
33 |     CONF_FILE = False
34 |     DISABLE_ACTIONS_WHEN_HIDDEN = False
35 |
36 |     # --- SpyderDockablePlugin API
37 |     #
38 |     def get_name(self):
39 |         return _('Plots')
40 |
41 |     def get_description(self):
42 |         return _('Display, explore and save console generated plots.')
43 |
44 |     def get_icon(self):
45 |         return self.create_icon('hist')
46 |
47 |     def register(self):
48 |         # Plugins
49 |         ipyconsole = self.get_plugin(Plugins.IPythonConsole)
50 |
51 |         # Signals
52 |         ipyconsole.sig_shellwidget_changed.connect(self.set_shellwidget)
53 |         ipyconsole.sig_shellwidget_created.connect(
54 |             self.add_shellwidget)
55 |         ipyconsole.sig_shellwidget_deleted.connect(
56 |             self.remove_shellwidget)
```

VS Code avec Jupyter Extension

The screenshot displays the VS Code Jupyter Notebook interface. The top toolbar includes options for Code, Markdown, Run All, Restart, Clear All Outputs, Variables, and Outline. The current environment is 'base (Python 3.12.2)'. The notebook contains three code cells:

```
[1] import numpy as np
import pandas as pd
```

```
[4] print("Version numpy est :"+np.__version__)
print("Version pandas est :"+pd.__version__)
```

...
Version numpy est :1.26.4
Version pandas est :2.2.3

```
[75] data = pd.read_csv("ventes_produits.csv")
```

The bottom panel shows the Terminal with the following output:

```
(base) bilal:projects bilal$
* History restored

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) bilal:projects bilal$
```

Les plateformes complètes de Data science

- Les plateformes complètes de Data Science sont des environnements intégrés qui regroupent tous les outils nécessaires pour réaliser l'ensemble du processus de data science, de la collecte des données jusqu'à la visualisation et le déploiement des modèles. Ces plateformes permettent aux data scientists de collaborer, d'automatiser des workflows, et de gérer des projets complexes en un seul endroit.

Anaconda

The screenshot displays the Anaconda Navigator interface, specifically the Learning section. The top navigation bar includes the Anaconda Navigator logo and a 'Connect' button. The left sidebar contains navigation options: Home, Environments, Learning (highlighted), and Community. The main content area features a grid of 24 documentation links, organized into two tabs: 'Documentation (26)' and 'Training (1)'. Each link includes a logo, a title, and a 'Read' button. The links are as follows:

Icon	Title	Action
Python logo	Python Tutorial	Read
Python logo	Python Reference	Read
Anaconda logo	Anaconda Package List	Read
pandas logo	Pandas Documentation	Read
Numpy logo	Numpy Documentation	Read
Scipy logo	Scipy Documentation	Read
Matplotlib logo	Matplotlib Documentation	Read
Bokeh logo	Bokeh User Guide	Read
Anaconda logo	Anaconda Cloud Documentation	Read
Anaconda logo	Anaconda Documentation	Read
Anaconda logo	Anaconda Navigator Documentation	Read
R logo	The Comprehensive R Archive Network (CRAN)	Read
Python logo	The Python Package Index (PyPI)	Read
Dask logo	Dask documentation	Read
Conda logo	Conda & Conda-Build	Read
Jupyter logo	Jupyter documentation	Read
Spyder logo	Spyder documentation	Read
VSCode logo	VSCode (python)	Read
Scikit-learn logo	scikit learn	Read
Orange3 logo	Orange3	Read
TF logo	TF	Read
PC logo	PC	Read
lab logo	lab	Read

Microsoft Azure

Platform Services

Security and Management

- Portal
- Active Directory
- Multi-Factor Authentication
- Automation
- Key Vault
- Store/Marketplace
- VM Image Gallery and VM Depot

Compute

- Cloud Services
- Service Fabric
- Batch
- Remote App

Web and mobile

- Web Apps
- API Apps
- API Management
- Mobile Apps
- Logic Apps
- Notification Hubs

Developer services

- Visual Studio
- Azure SDK
- Team Project
- Application Insights

Hybrid Operations

- Azure AD Connect Health
- AD Privileged Identity Management
- Backup
- Operational Insights
- Import/Export
- Site Recovery
- StorSimple

Integration

- Storage Queues
- BizTalk Services
- Hybrid Connections
- Service Bus

Analytics and IoT

- HDInsight
- Machine Learning
- Data Factory
- Event Hubs
- Stream Analytics
- Mobile Engagement

Data

- SQL Database
- SQL Data Warehouse
- Redis Cache
- Search
- Cosmos DB
- Tables

Media and CDN

- Media Services
- Content Delivery Network (CDN)

Infrastructure Services

Compute

- Virtual Machine
- Containers

Storage

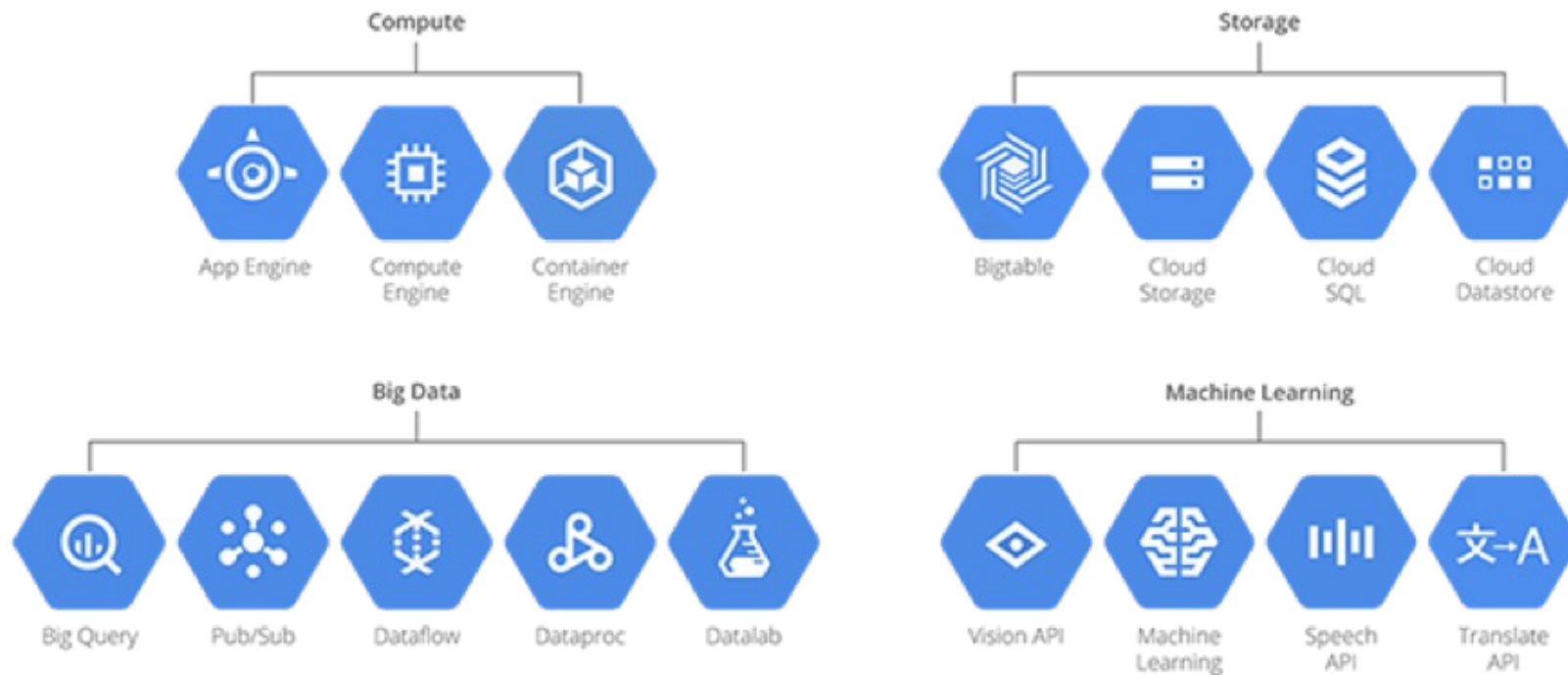
- BLOB Storage
- Azure Files
- Premium Storage

Networking

- Virtual Network
- Load Balancer
- DNS
- Express Route
- Traffic Manager
- VPN Gateway
- Application Gateway

Google cloud platform

Google Cloud Platform



Data science on AWS

