



**UNIVERSITE ABDELMALEK ESSAADI**

**FACULTE DES SCIENCES ET TECHNIQUES DE TANGER**

**DEPARTEMENT GENIE INFORMATIQUE**

**PROJET DE FIN D’Module**

**Master Mobiquité & Big Data**

**« Rapport de projet de fin de module NoSQL**

**(Cryptocurrency Data Visualisation / Analysis) »**

**Réalisé PAR :**

**Benamar Zaid**

**Sous l’encadrement de :**

**Pr. M. EL AACHAK Lotfi**



**Fais-le : 01/06/2022**

***Année Universitaire : 2021/2022***

**TABLE DES MATIÈRES :**

Partie I : Contexte générale du projet

INTRODUCTION……………………………………………………………………………..1

1. Objectif du projet ……………………………………………………………...2
2. Technologies et outils utilisé ………………………………………………......2

II.1. Front-end ……………………………………………………………..2

II.2. Back-end ……………………………………………………………...3

II.3. Analyse des données ………………………………………………….3

II.4. Web Scraping ………………………………………………………...4

II.5. Stockage des données ………………………………………………...4

Partie II : Préparation et analyse des données

1. Collection des données …………………………………………………………..5
2. Prétraitement des données ……………………………………………………….7
3. Chargement des données dans MongoDB ……………………………………….13
4. Analyse des données ……………………………………………………………..17

IV.1. Série de retour ………………………………………………………….18

IV.2. Volatilité ………………………………………………………………..19

IV.3. Rendements cumulatifs ………………………………………………...20

IV.4. Corrélation ……………………………………………………………..21

Partie III : Réalisation

1. Présentation des interfaces du travail réalisé ……………………………………..23

CONCLUSION…………………………………………………...……………………..……26

**INTRODUCTION :**

L'analyse des données est la méthode permettant d'extraire des solutions aux problèmes par l'interrogation et l'interprétation des données. Le processus d'analyse comprend la découverte des problèmes, la résolution de l'accessibilité des données appropriées, la détermination de la méthode qui peut aider à trouver la solution au problème intéressant et la transmission du résultat. Pour les besoins de l'analyse, les données doivent être séparées en plusieurs étapes telles que la spécification, l'assemblage, l'organisation, le nettoyage, la ré-analyse, l'application de modèles et d'algorithmes et le résultat final. Un nombre considérable d'individus ont utilisé ces stratégies dans la recherche et les affaires pour créer de la substance ou offrir des critiques pour étendre l'exactitude de la publicité commerciale qui permet aux individus de fournir des ressources dans l'avancement et le développement de l'entreprise.

**PARTIE I**

**« Contexte générale du projet »**

1. **Objectif du projet**

L’objectif principal du projet est la réalisation d’une tableau de bord de type single page application afin de visualisé les analyses des données qui sont stocké dans une base de données NoSQL orienté document, Il s’agit des données historiques sur 4 **crypto-monnaies (Cryptocurrencies)** extracté à partir le Scrapping qui les collecte d’après le site officiel <https://coinmarketcap.com/>.

1. **Technologies et outils utilisé**

Afin de réaliser ce Project, on a utilisé plusieurs technologies lesquelles :

1. **Front-end :**

**Angular 8 :** Angular est une plate-forme et un framework permettant de créer des applications clientes de type single-page application à l'aide de HTML et de TypeScript. Angular est utilisé pour créer des applications Web dynamiques.

**Plotly.js :** Plotly.js est une bibliothèque de graphiques JavaScript qui comprend plus de 40 types de graphiques, des graphiques en 3D, des graphiques statistiques et des cartes SVG.

1. **Back-end :**

**Django** **:** Django est un framework web python open-source utilisé pour le développement rapide, pragmatique, maintenable, propre, et sécurise les sites web. Participe au développement Web permet aux utilisateurs de se concentrer sur le développement des composantes nécessaires à leur application.

**Djongo** : Djongo est une approche unifiée de l'interfaçage des bases de données. Il s'agit d'un prolongement du Django ORM framework mais mappe les objets python aux documents MongoDB.

**Django REST Framework :** Django REST framework (DRF) est une bibliothèque Python/Django open source, mature et bien prise en charge qui vise à créer des API Web sophistiquées. Il s'agit d'une boîte à outils flexible et complète avec une architecture modulaire et personnalisable qui permet le développement de points de terminaison d'API simples et clés en main et de constructions REST complexes**.**

**Elasticsearch :** Elasticsearch est un moteur de recherche et d'analyse distribué gratuit et ouvert pour tout type de données, la vitesse et la scalabilité d'Elasticsearch, ainsi que sa capacité à indexer de nombreux types de contenus signifient qu'il peut être employé dans différents cas d'utilisation.

1. **Analyse des données :**

**Jupyter Notebook  :** Jupyter Notebook est une application Web open source que vous pouvez utiliser pour créer et partager des documents contenant du code en direct, des équations, des visualisations et du texte.

**Pandas :** Pandas est un paquetage Python open source très largement utilisé pour la science des données, l'analyse des données et les tâches d'apprentissage automatique. Il vise à être le composant fondamental de haut niveau pour faire une analyse de données pratique et réelle en Python.

**Seaborn** **:** Seaborn est une bibliothèque de visualisation de données pour le traçage de graphiques statistiques en Python. Il fournit de beaux styles par défaut et des palettes de couleurs pour rendre les tracés statistiques plus attrayants. J’ai utilisé bibliothéque pour visualisé les données lors de l’analyse dans jupyter notebook.

**Matplotlib & NumPy** : Matplotlib est une bibliothèque de traçage pour Python. Il est utilisé avec NumPy pour fournir un environnement qui est une alternative open source efficace.

1. **Web Scrapping :**

**Beautiful Soup** **:** Beautiful Soup est un package Python pour l'analyse de documents HTML et XML. Il crée un arbre d'analyse pour les pages analysées qui peuvent être utilisées pour extraire des données de HTML, ce qui est utile pour le grattage Web.

1. **Stockage des données :**

**MongoDB :** MongoDB est une base de données orientée documents. En clair, vous bénéficiez de la scalabilité et de la flexibilité que vous voulez, avec les fonctions d’interrogation et d’indexation qu’il vous faut.

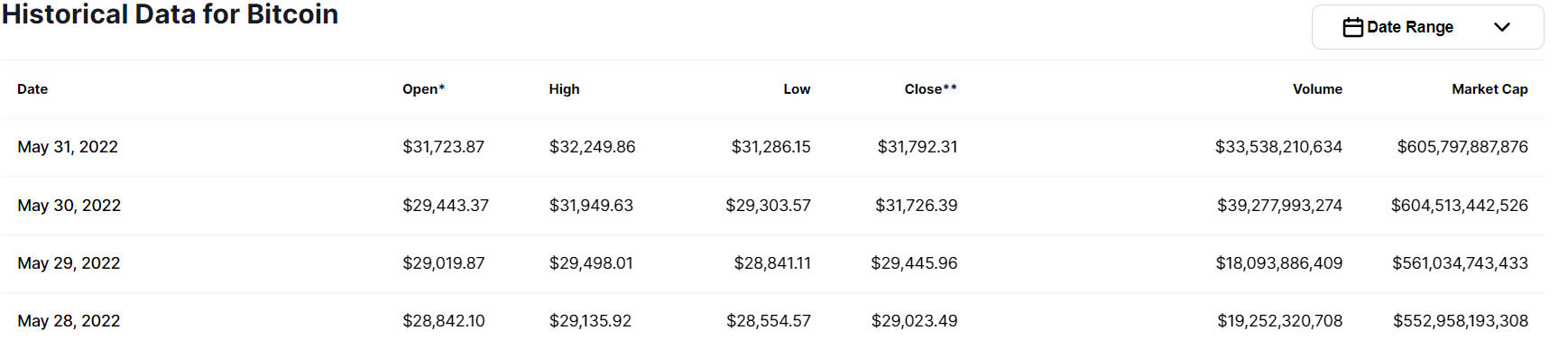
**PARTIE II**

**« Préparation et analyse des données »**

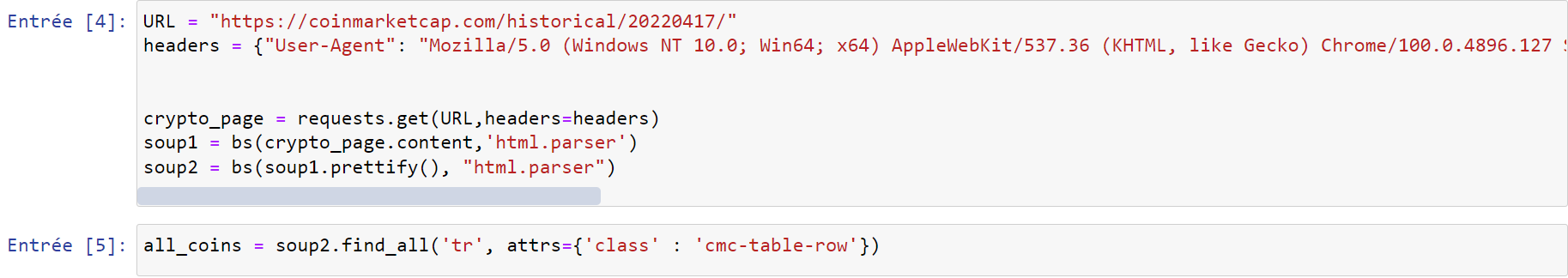
1. **Collection des données**

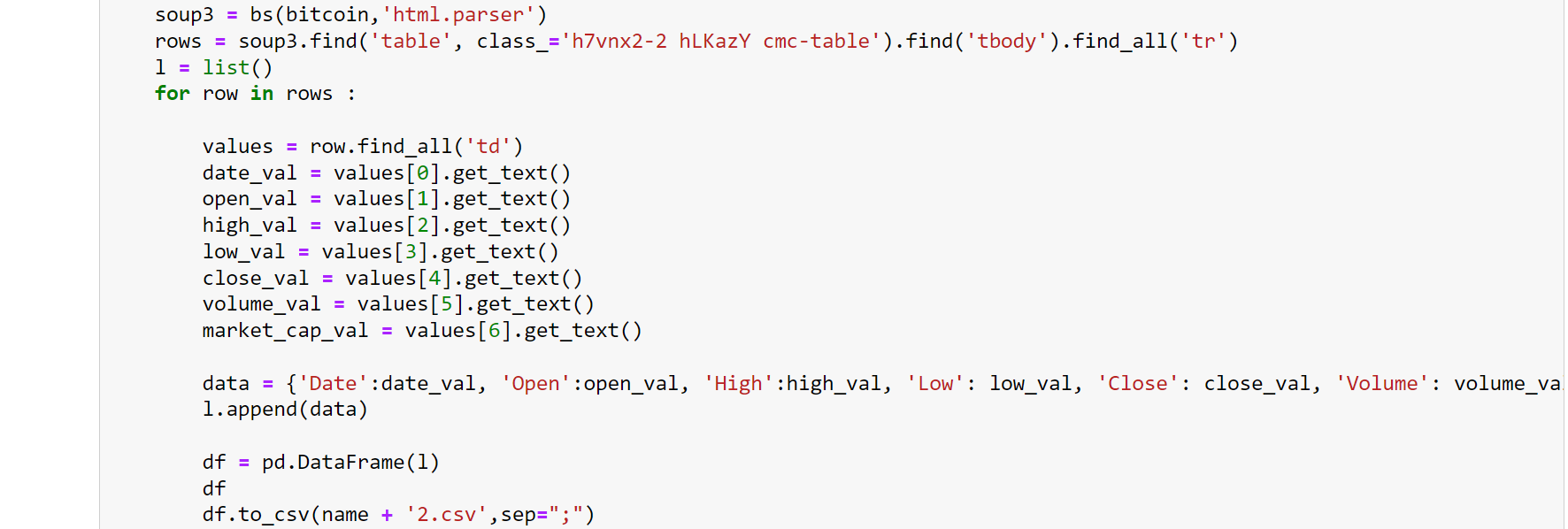
Afin de récupérer les données depuis **CoinMarket** sur lesquelles on va travailler. On a utilisé l’outil **Beautiful Soup** qui offre plusieurs mécanismes d’extraire les données depuis les pages Web, comme **find** qui permet de sélectionner les bloc de code HTML on spécifiant le nom de la balise.

Par exemple, afin de récupérer les données historiques de Bitcoin depuis **CoinMarket.**



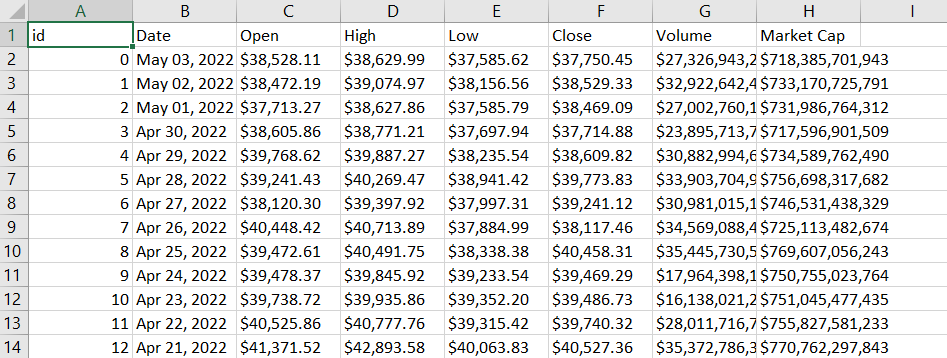
On peut procéder comme suit :





Le processus derniér serve a extracté les données **d’une durée de 1 an** de chaque crypto et les stocké dans une Dataframe afin de l’exporté dans un fichier csv pour l’analyse.

* **Résultat :**



**Date :** date de l'observation

**Open :** Le prix d'ouverture

**High :** Le prix le plus élevé

**Low :** Prix le plus bas

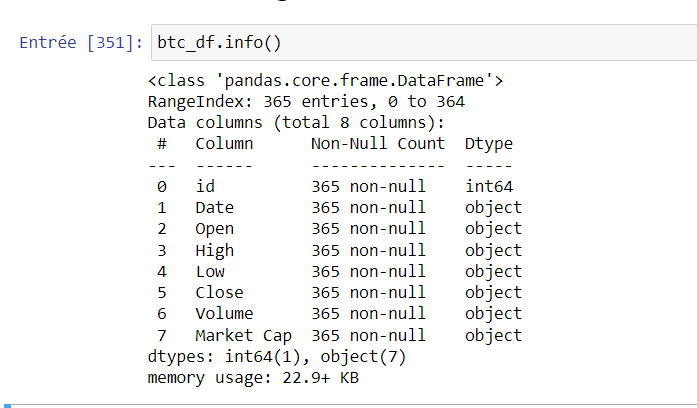
**Close :** Le prix de clôture

**Volume :** Total des actions négociées

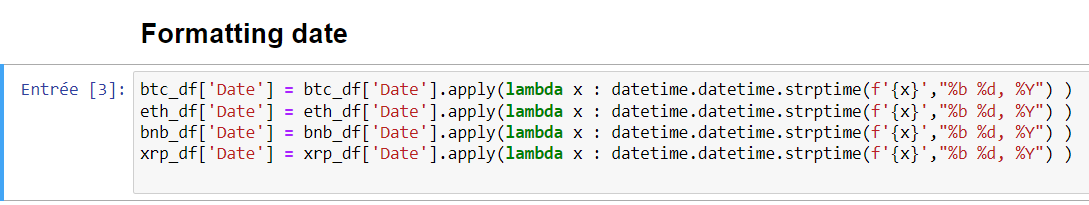
**Market Cap :** La valeur totale de toutes les pièces qui ont été frappées

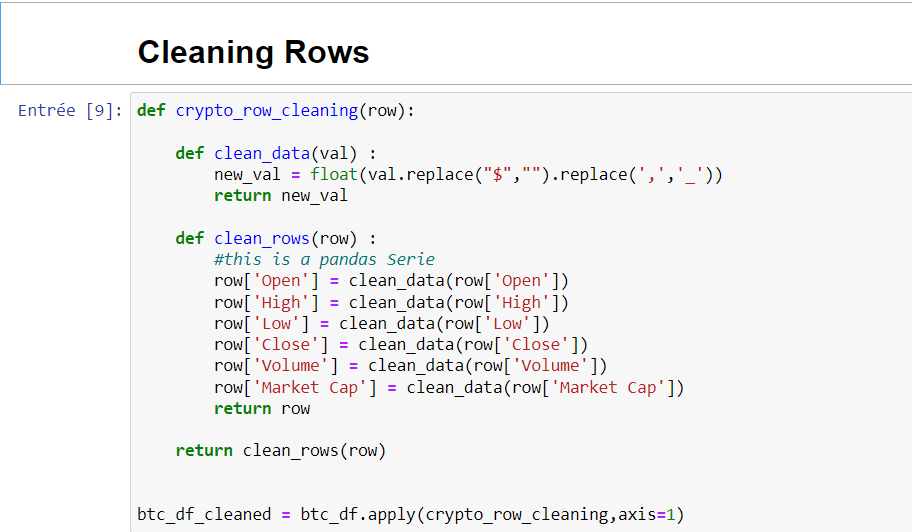
1. **Prétraitement des données**

La premiére des choses avant de procédé au processus de detection des missing values et gérer les valeurs aberrantes ( outliers ), on va tout d’abord préparer le dataframe pour cela.

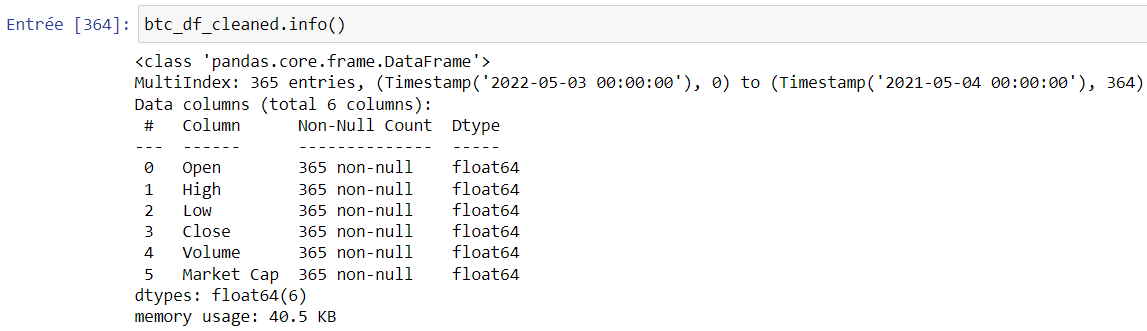


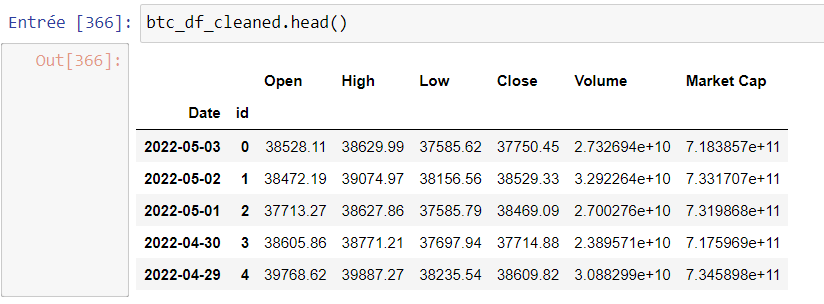
Tout les champ sont de type object (String), ce qu’est pas compatible avec l’analyse. Concernant le champ Date on va le formatter pour qu’il devient de type date, et les autres champs en les convertissant en nombres entiers.





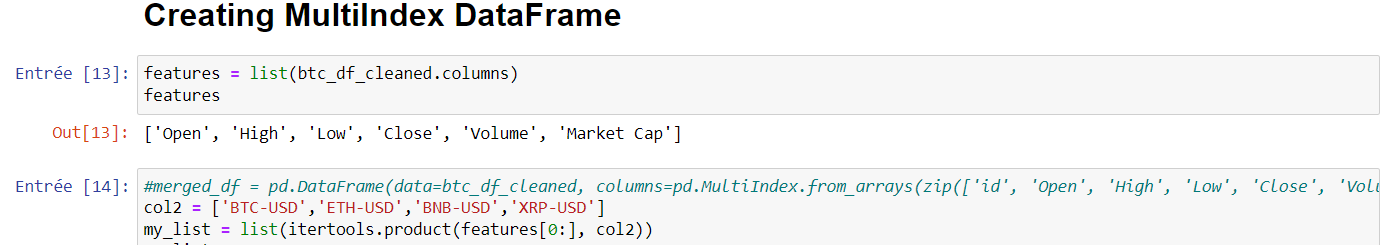
* **Résultat :**



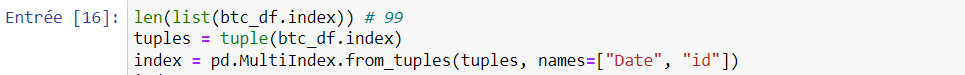


Jusqu’à moment nous disposont de 4 dataframes, chaqune contient des données sur un crypto monnaies spécifique, et pour chaqune on a effectuer le processus de formatage précédant.

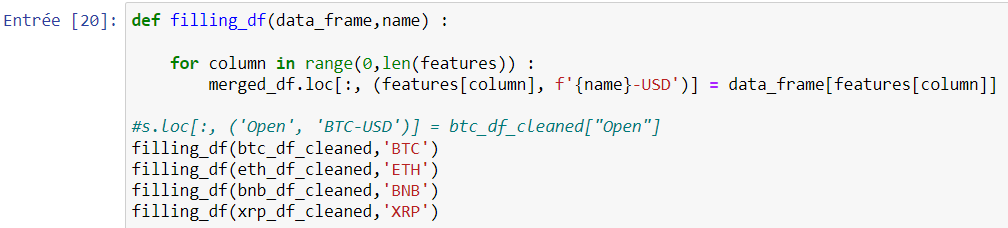
On veut d’abord combiner ces 4 dataframe pour qu’ils devient une seul, cela nécéssite la création d’une MultiIndex Dataframe, donc le processus suivante vise a atteint ce objectif.



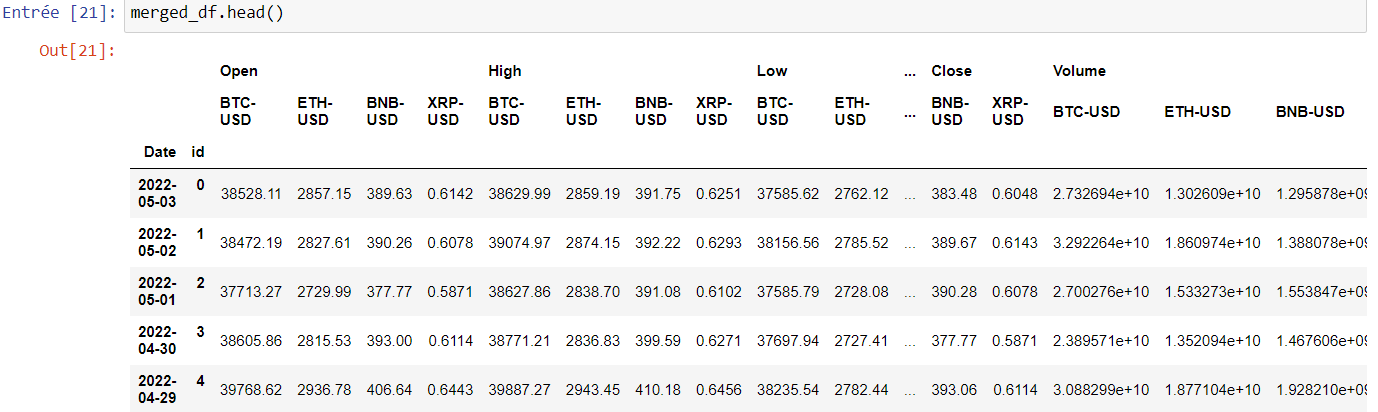




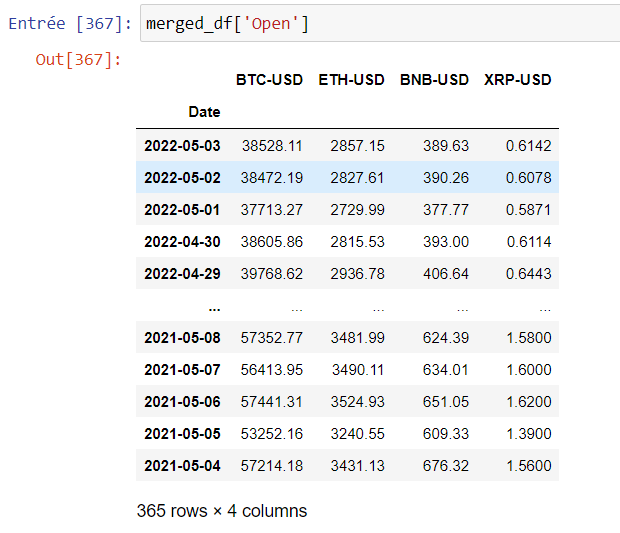
La méthode **filling\_df** vise à remplire la nouvelle dataframe avec les données des 4 autre dataframes.



* **Résultat :**

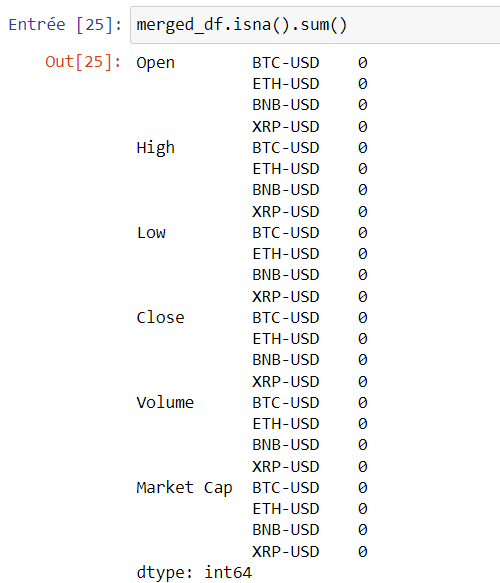
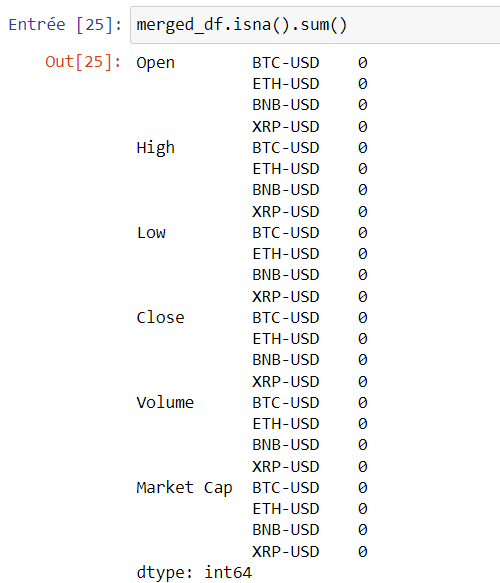


Avec la nouvelle dataframe qu’on a crée on peut mieux visualisé les données des 4 crypto monnaies par rapport a l’un des features comme suit :

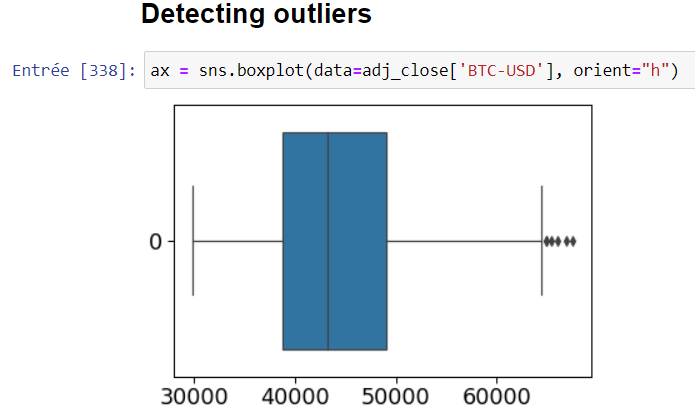


Il nous reste d’abord que la detection des missing values et gérer les valeurs aberrantes ( outliers ).

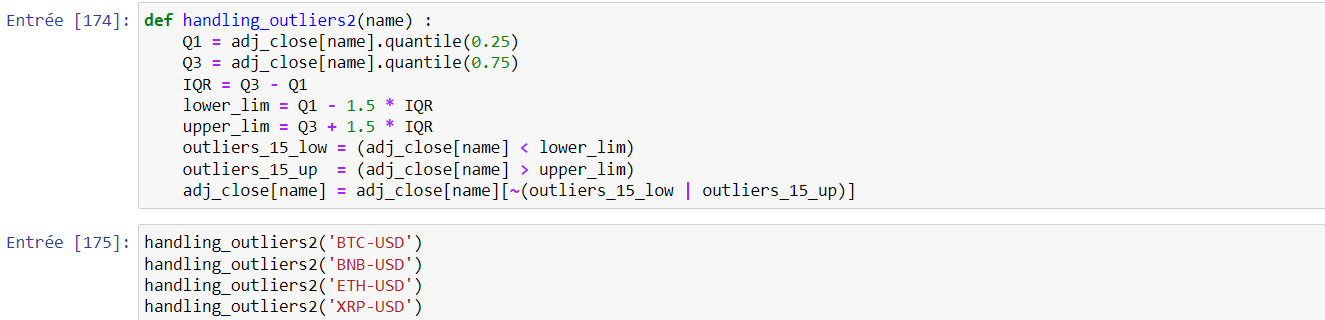
Cocernant les valeurs manquants, notre MultiIndex dataframe contient aucun.



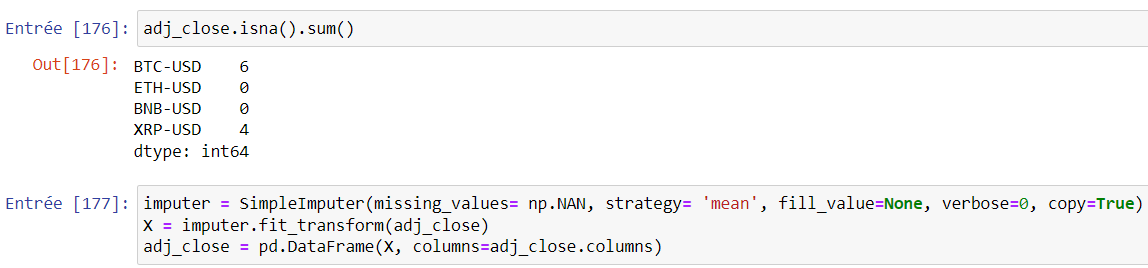
D’abord on va voir si notre dataframe contient des outliers à l’aide d’un boxplot, et puisque nous sommes intéressés par le prix de clôture (Closing Price) de chaque crypto monnaies, donc on va traité les outliers just pour ces colonnes.



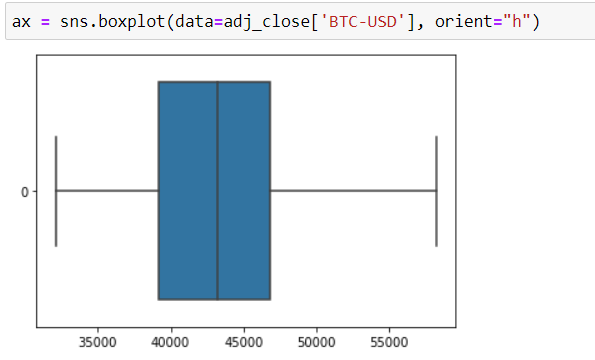
On remarque par exemple que la colonne « BTC-USD » qui référence au Bitcoin ayant des outliers, pour les enlevé on va faire le jeux d’opérations suivant :



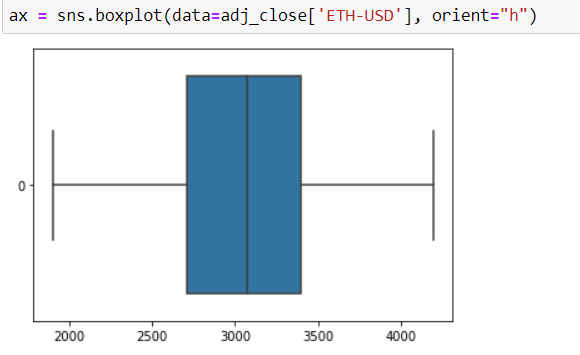
Après ce traitement on remarque que le dataframe contient des missing values, donc j’ai essayé de les enlevé apartir cette méthode :

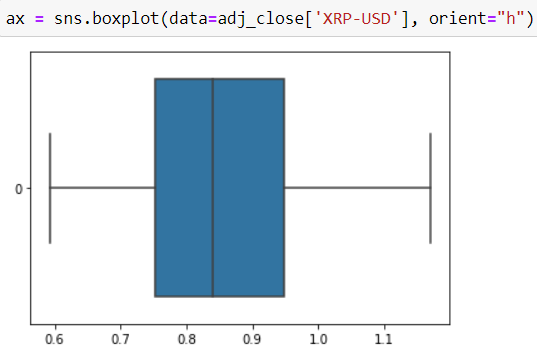


* **Résultat de Bitcoin :**



* **Résultat de Ethereum :**

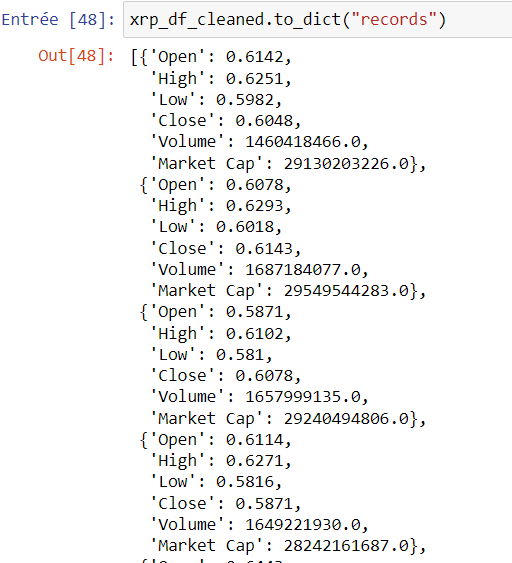


* **Résultat de XRP :**

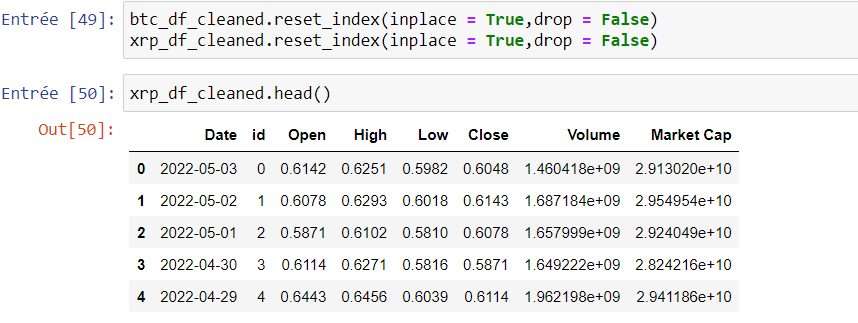
1. **Chargement des données dans MongoDB**

Après avoir finir l’étape de pre-processing, on a besoins de stocker ces données dans une bases de données pour les utilisées.

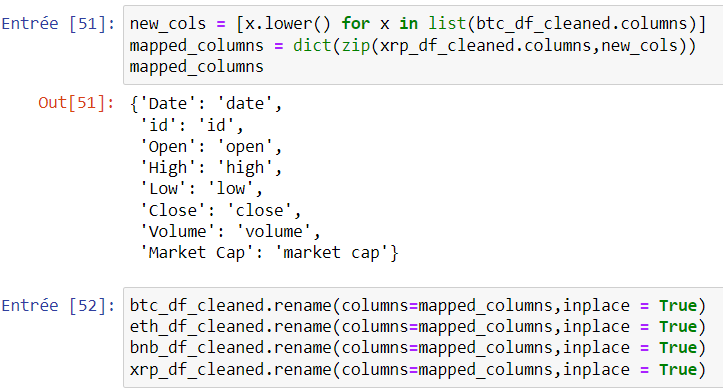
La méthode to\_dict() de Pandas convertie une dataframe au dictionnaire, ce qui est équivalent au format des documents JSON au base de données MongoDB**.**

****

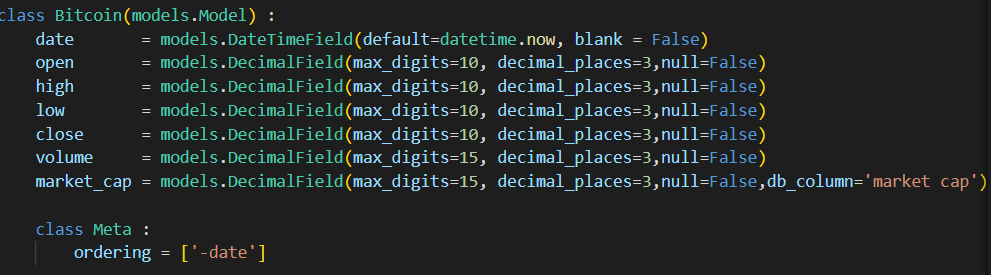
Il faut aussi réinitialiser l'index qui est la Date, sinon il ne va pas être stocké lors de chargement de dataframe.



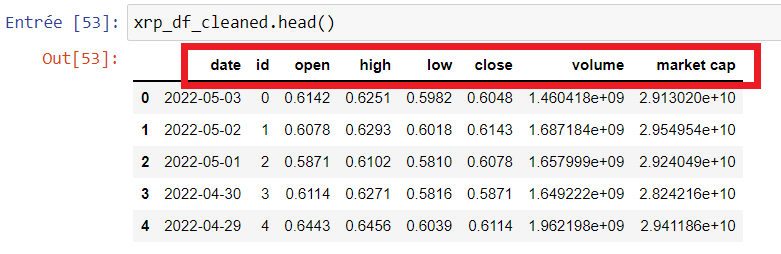
Aussi j’ai fait un étape très important qu’est de renommer les colonnes de dataframe pour qu’il devient adapté au modèle créer pour celle-là dans Django.



Les noms des colonnes définie dans Django pour tout les crypto monnaies sont les suivants :



* **Résultat :**

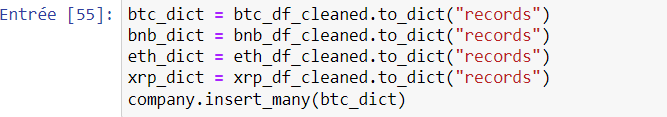


Le dernier étape maintenant et de chargé ce dataframe dans MongoDB, cela nécessite l’utilisation de bibliothéque **PyMongo** pour se connecte et interagit avec les bases de données.

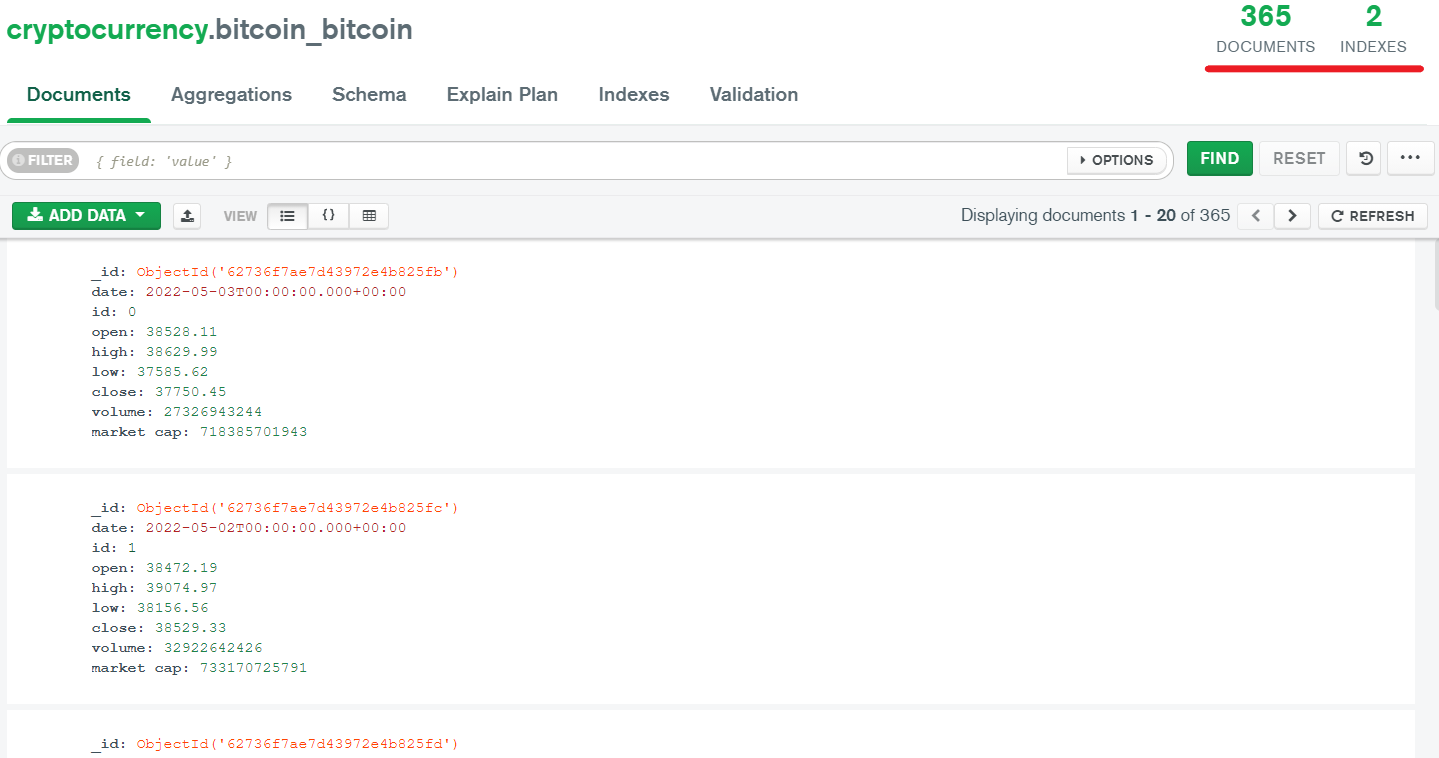
Tout d’abord il faut établir une connection vers la base de données, on spécifiant à la fois le nom de la BDD et la collection où on souhaiter de stocker les donneés.



On peut d’abord chargé le dataframe dans notre collection comme suite :

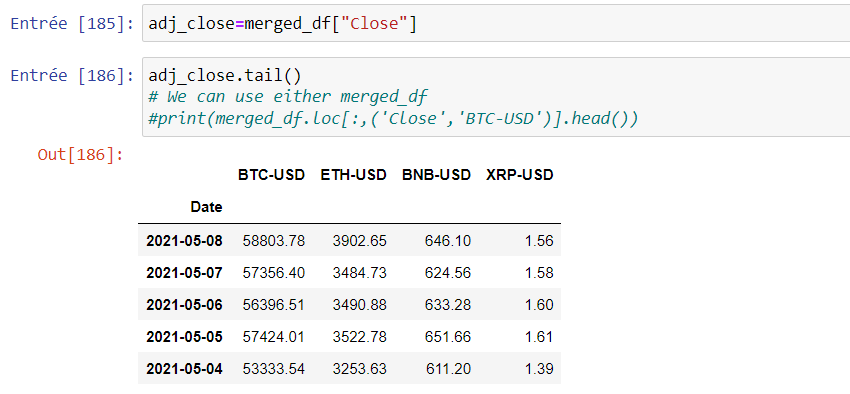


* **Résultat :**

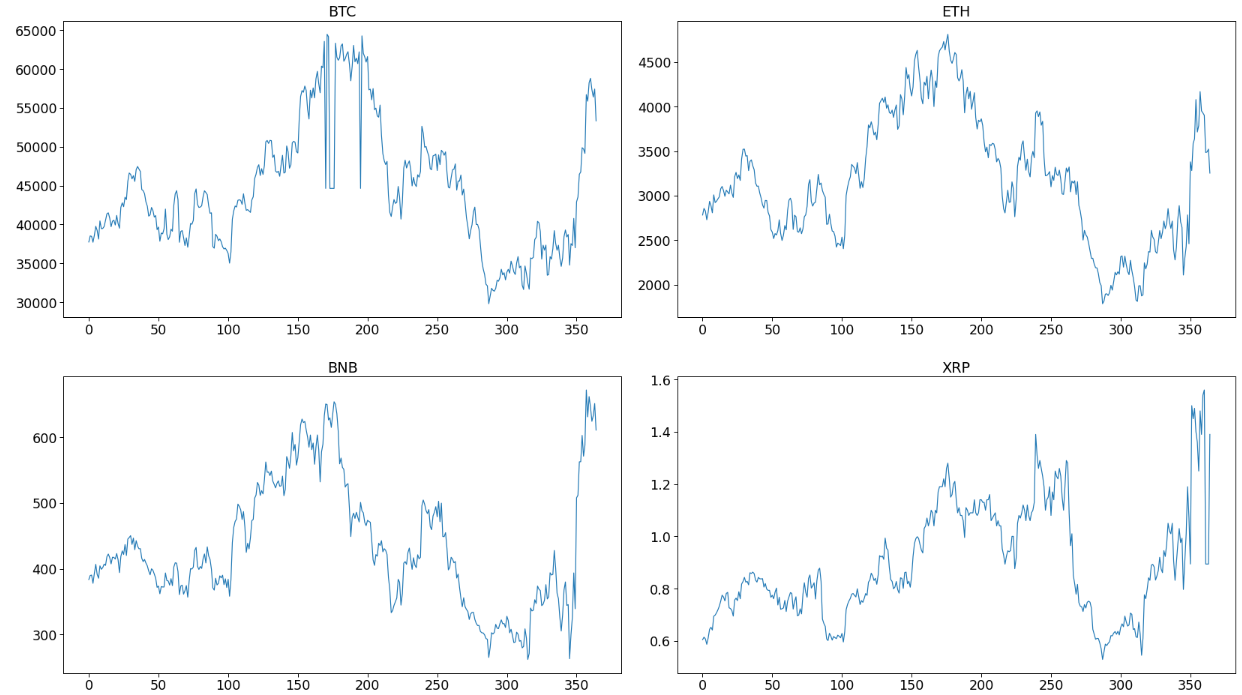


1. **Analyse des données**

Nous sommes intéressés par le Adjusted closing price. Par conséquent, nous allons sélectionner le prix de clôture ajusté des crypto-monnaies.



Nous allons tracer le Adjusted closing price. Nous utilisons des subplots puisque les crypto-monnaies sont sur des échelles différentes.



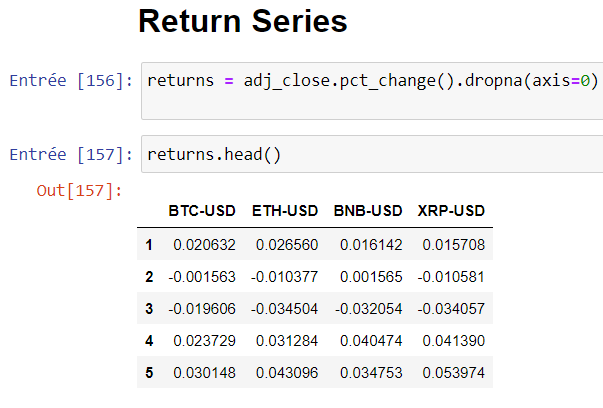
Les graphiques sont des graphiques de séries chronologiques ( time series plots ), indiquant l'évolution du cours de l'action dans le temps. Comme les échelles sont différentes, nous ne pouvons pas comparer les graphiques, mais nous utiliserons plutôt le graphique des rendements cumulés.

1. **Série de routeur**

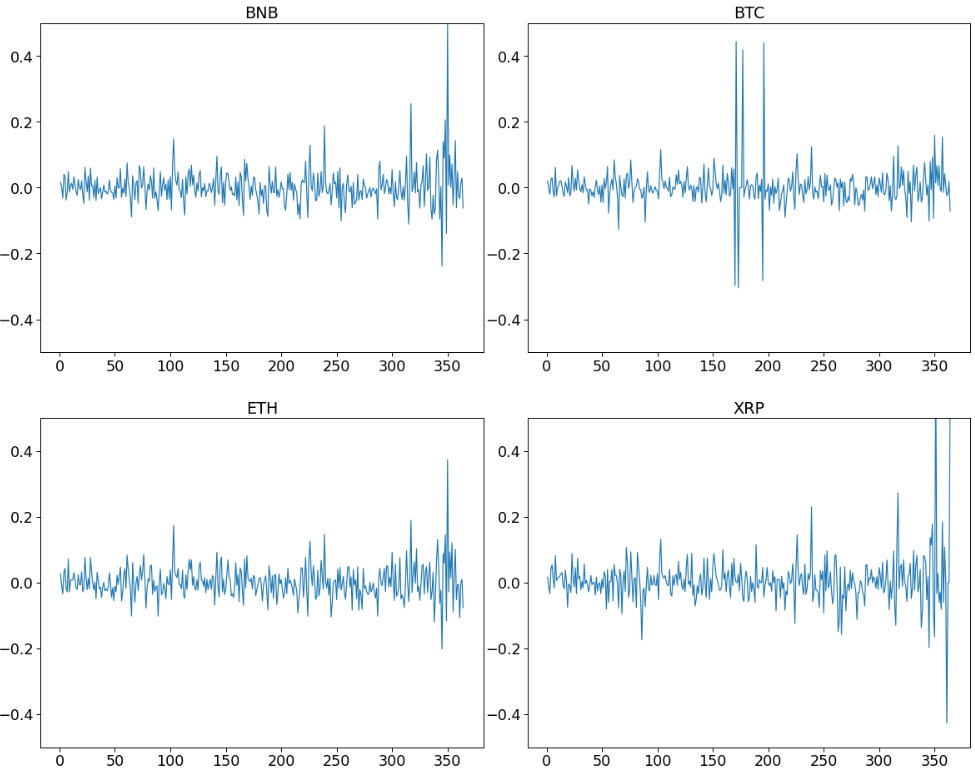
Un rendement (return) est une variation du prix d'un actif dans le temps.

Les rendements peuvent être positifs, représentant un profit, ou négatifs, indiquant une perte.

Nous allons utiliser la fonction pandas pct\_change() pour calculer les rendements.



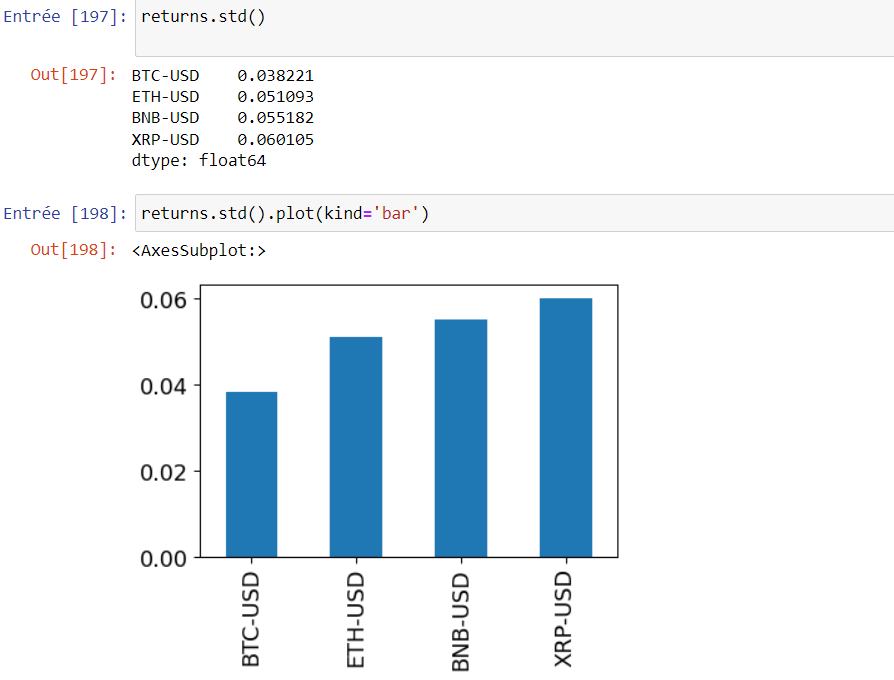
Voici les changements de prix relatifs des crypto-monnaies.

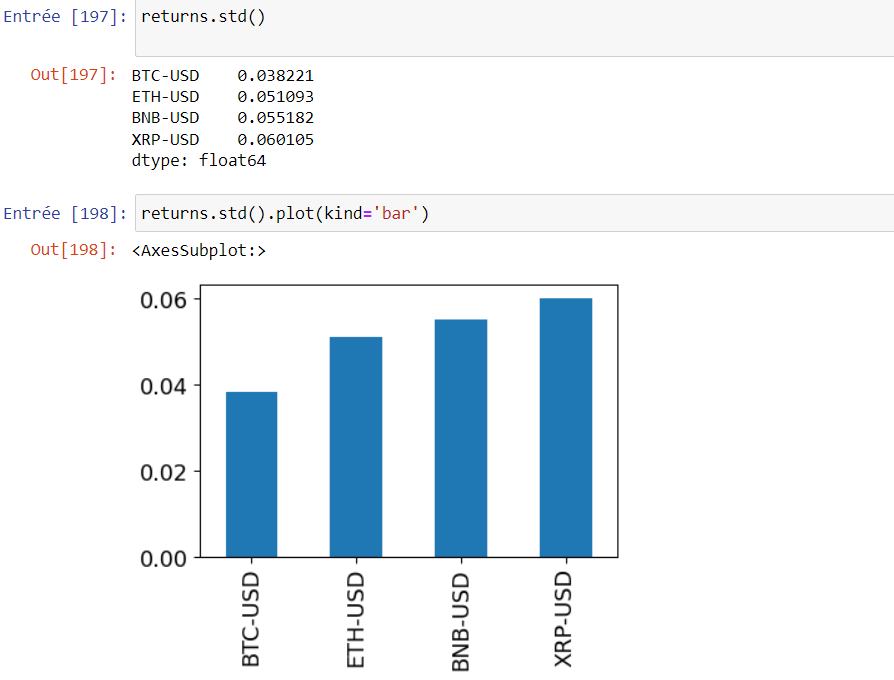


XRP est la plus volatile, suivie de Bitcoin, tandis que Ethereum est la moins volatile.

1. **Volatilité**

La volatilité est une mesure de la variation du prix d'un actif dans le temps. Plus l'écart-type est élevé, plus un actif est volatil.



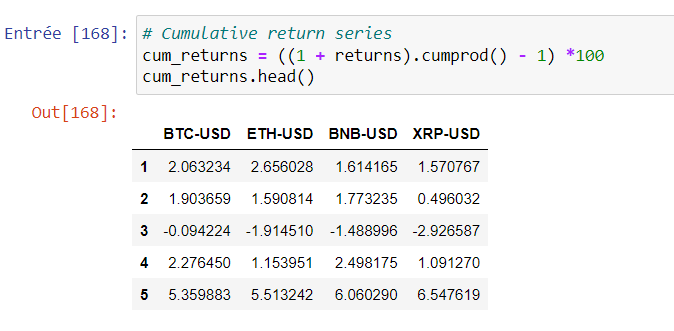


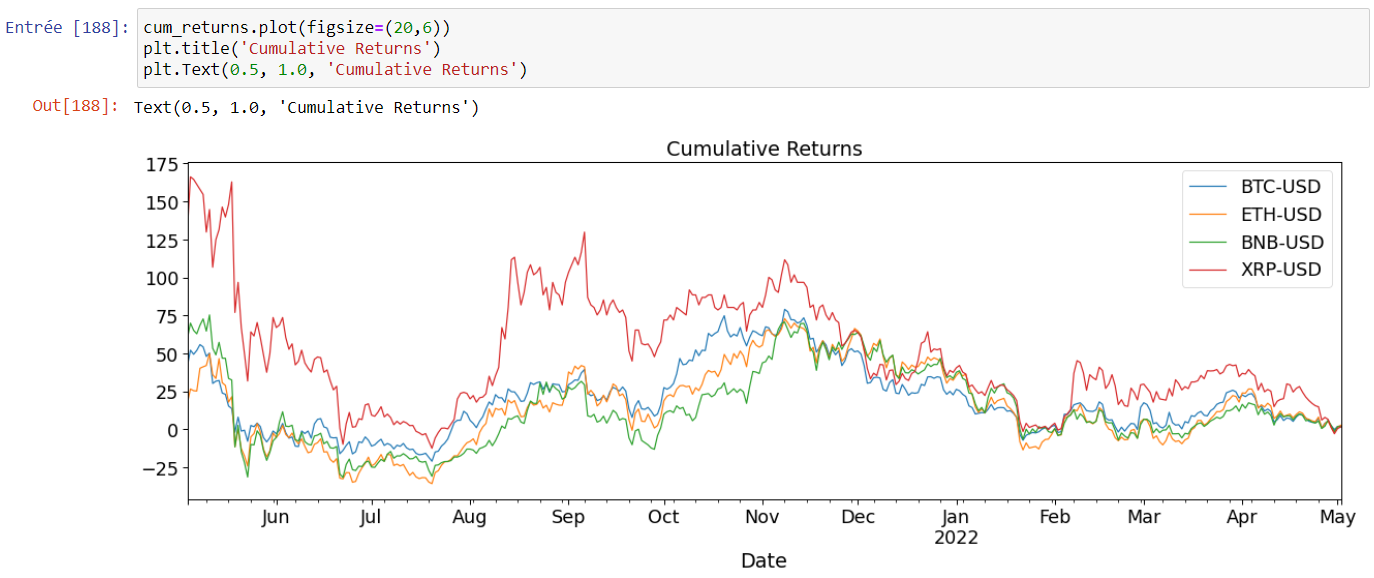
XRP (Ripple) est le plus volatil des quatre actifs, tandis que le Bitcoin est le moins volatil.

1. **Rendements cumulatifs**

Le rendement cumulé exprime la variation totale du prix d'un actif dans le temps.

Nous utilisons la fonction pandas cumprod() pour calculer les rendements simples cumulés quotidiens.



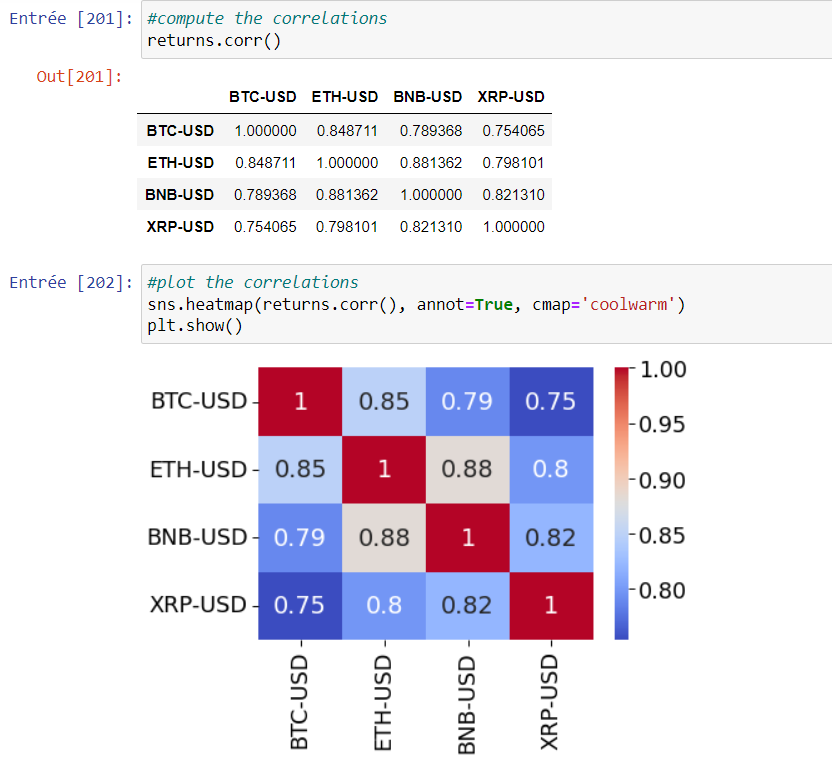


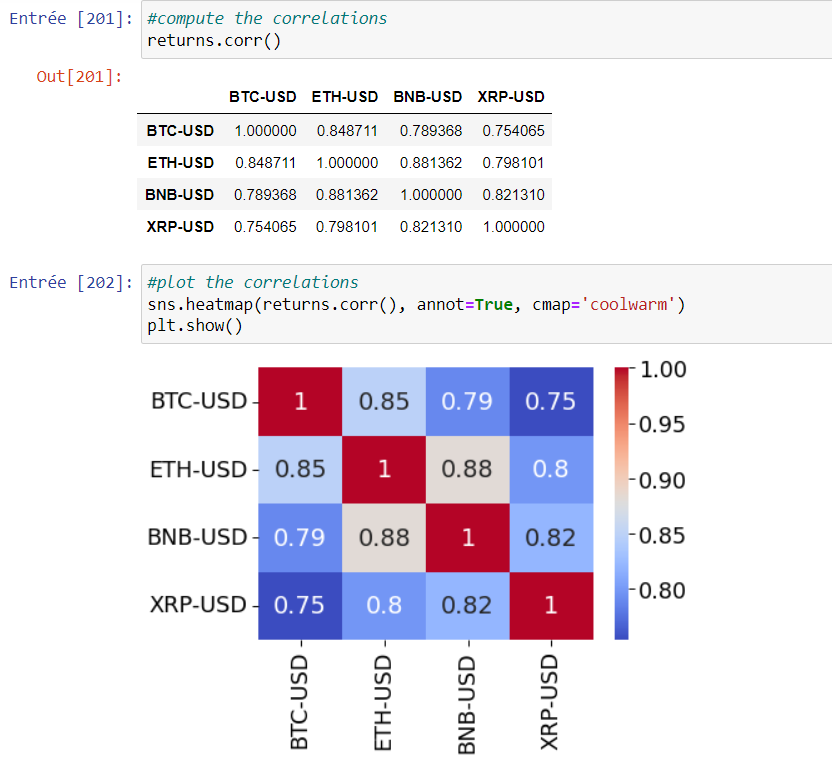
XRP (Ripple) surpasse l'Ethereum, le Bitcoin et le Binance à partir de juin 2021, tandis que le Bitcoin surpasse l’Ethereum et le XRP.

1. **Corrélation**

J’ai calculé la corrélation sur le rendement car la corrélation sur les données de prix brutes peut donner des résultats biaisés.

Les coefficients de corrélation proches de 1 indiquent une forte association positive, -1 indique une forte association négative et les coefficients proches de zéro n'indiquent aucune association.





Nous pouvons voir que le Binance et l'Ethereum sont fortement corrélés. Cela signifie que lorsque le Binance augmente, l'Ethereum augmente également, et lorsque le Binance diminue, l'Ethereum diminue aussi.

**PARTIE III**

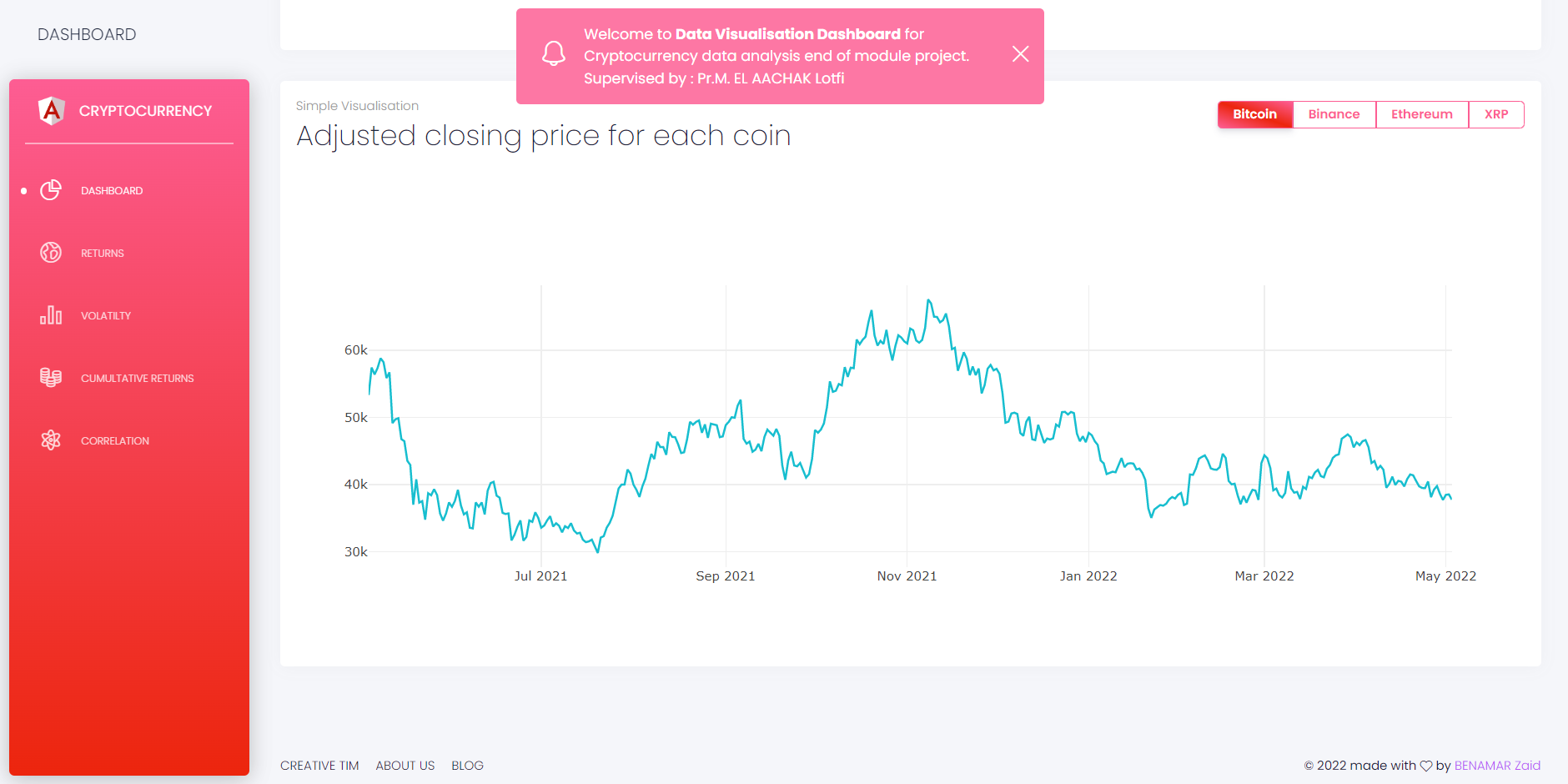
**« Réalisation »**

1. **Présentation des interfaces de travail réalisé**

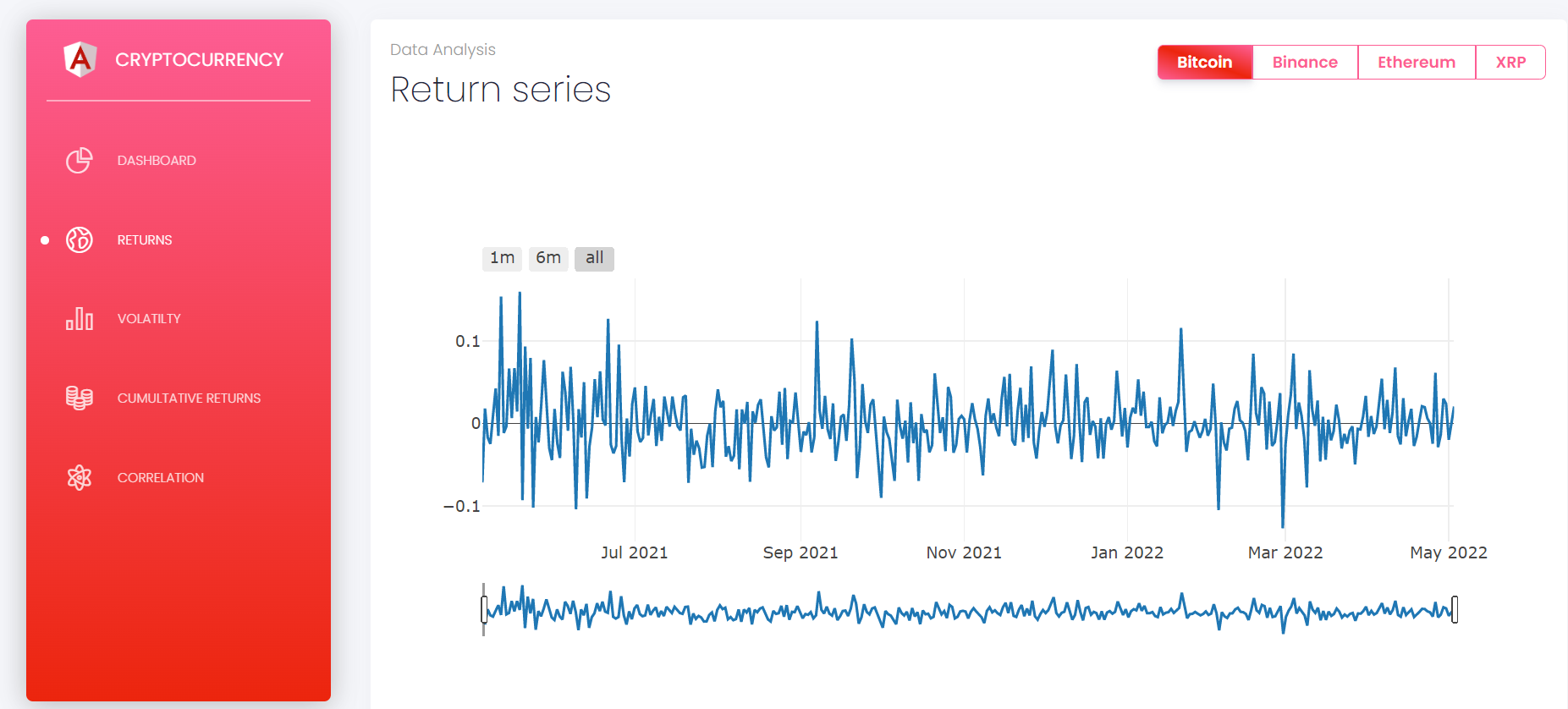
Dans cette partie on va réalisé une tableau de bord **(Dashboard)** de type single page application afin de visualisé les analyses des données qu’on a effectué dans la partie précédente.

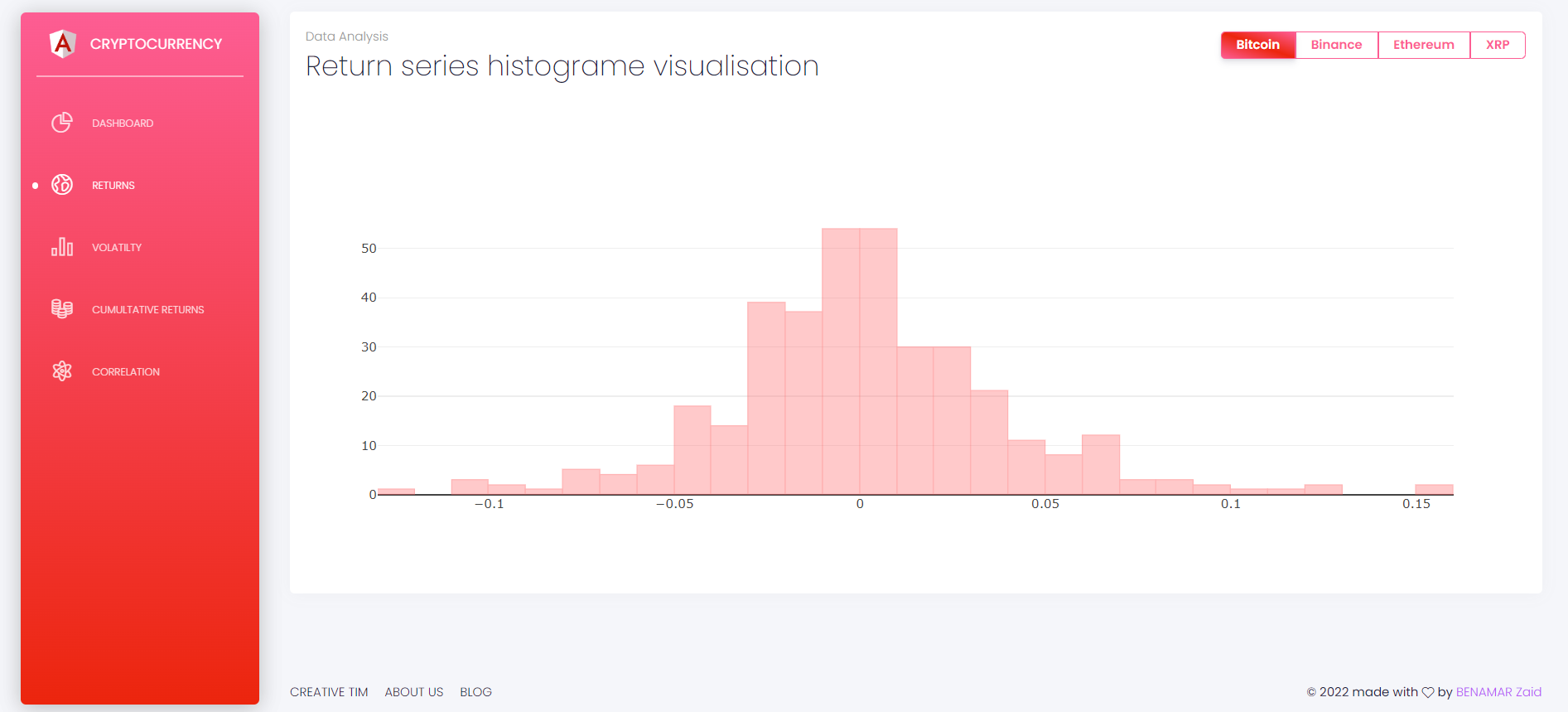
Le tableau de bord developpé par Angular 13 qui contient des graph affiché par la bibliothéque Plotly.js.



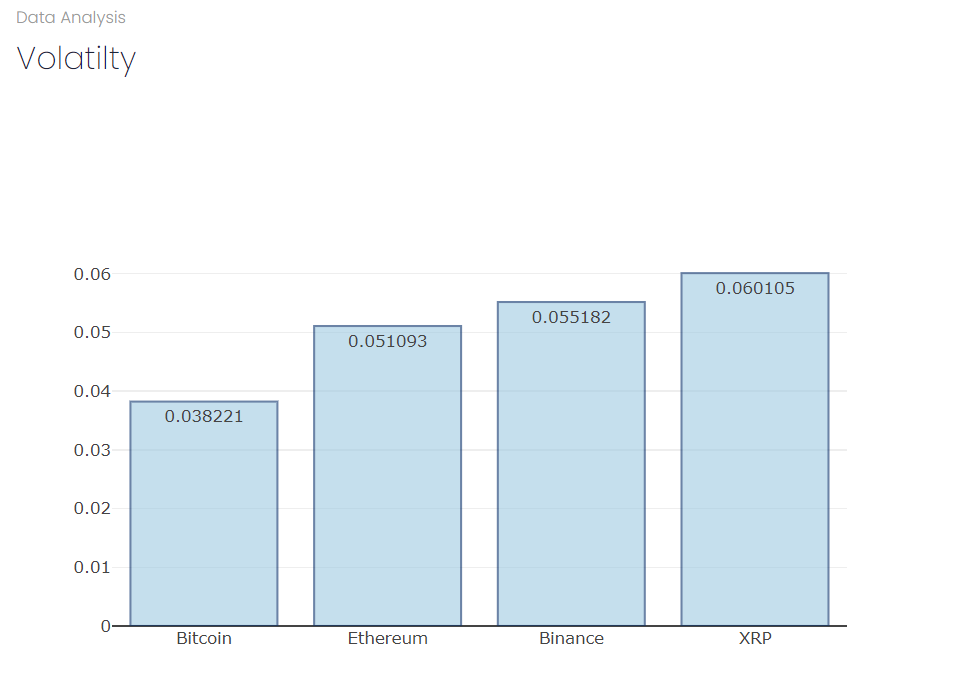


1. **Série de routeur**





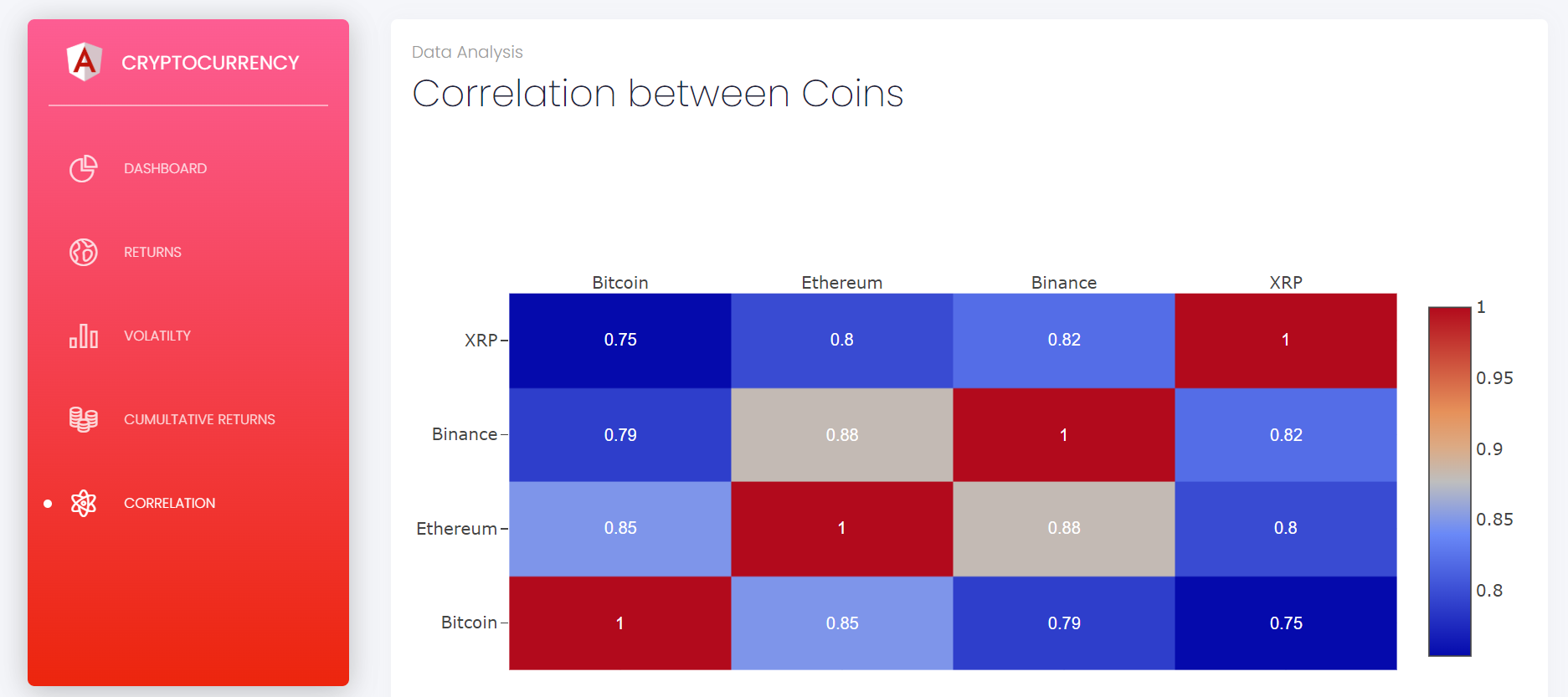
1. **Volatilité**



1. **Rendements cumulatifs**



1. **Corrélation**



**CONCLUSION :**

Ce travail a pour objectif de concevoir et faire face aux différentes techniques de la préparation et l’analyse des données, qui permet d’évaluer les données à l'aide d'outils analytiques et statistiques.

Pour pouvoir compléter mon projet, j’avais détaillé les différentes étapes collection, de prétraitement et l’analyse de cette Data frame.

Durant ces parties, j’avais présenté la conception détaillée, à la fois, de mon projet, d’autre part j’avais aussi présenté les tâches qui a été accompli pour chaque partie, ensuite, j’ai présenté quelques captures d’écran montrant le bon fonctionnement du tableau de bord.