

Mise en place d’une infrastructure

Cryptomonnaie

Version 0.1



Table des matières

[I. Contexte du fil rouge 2](#_Toc91266698)

[A. Problématique 2](#_Toc91266699)

[B. Objectifs pour la v 0.1 3](#_Toc91266700)

[C. Principaux indicateurs clés 3](#_Toc91266701)

[D. Conceptualisation et Représentation 4](#_Toc91266702)

[II. Ingestion 5](#_Toc91266703)

[A. Gestion de projet et travail collaboratif 5](#_Toc91266704)

[B. Installation de la base de données 6](#_Toc91266705)

[C. Conceptualisation des données dans ES 7](#_Toc91266706)

[D. Ingestion via Logstash 8](#_Toc91266707)

[E. Ingestion et visualisation des données en temps réel via Python 13](#_Toc91266708)

[F. Tests de connectivité Power BI et Jupyter. 17](#_Toc91266709)

[III. Analyses et visualisation 20](#_Toc91266710)

[A. Visualisation via Kibana 20](#_Toc91266711)

[B. Visualisation et analyses via Power BI 21](#_Toc91266712)

[C. Visualisation et analyses via Jupyter et Pandas 26](#_Toc91266713)

[1. Analyse de la Capitalisation boursière 26](#_Toc91266714)

[2. Analyse des prix 29](#_Toc91266715)

[3. Analyse de la volatilité 31](#_Toc91266716)

[4. Conclusion de l’analyse 33](#_Toc91266717)

[IV. Conclusion générale 34](#_Toc91266718)

[A. Retour sur le projet 34](#_Toc91266719)

[B. Améliorations à venir pour la v 0.2 et ultérieurs 34](#_Toc91266720)



# Contexte du fil rouge

## Problématique

Notre entreprise réalise des profits importants générant une trésorerie conséquente. Une partie de cette trésorerie est placée en actifs financiers classiques (actions, obligations, ETF, OPCVM). Cependant votre direction financière désire diversifier ses actifs en recherchant de meilleures rentabilités.

Depuis plusieurs années : de grandes institutions bancaires, comme JP Morgan, souligne l’intérêt et l’opportunité des crypto-actifs comme le bitcoin, l’etherum ou cardano (<https://markets.businessinsider.com/news/currencies/ethereum-ether-bitcoin-crypto-defi-nfts-interest-rates-gold-jpmorgan-2021-11>). De grands cabinets comptables (KPMG/ PWC/ Deloitte) ont ouverts eux aussi des pôles dédiés aux crypto-actifs afin de sensibiliser leurs clients à ce nouveau type de produits.(<https://home.kpmg/fr/fr/home/insights/2021/06/cryptoactifs-vers-une-revolution-financiere.html>).

Il vous est demandé des indicateurs de performance pertinents pour de l’investissement ainsi qu’une étude décrivant les différents éléments clés du secteur. Vous disposez d’une carte blanche pour la mise en place de la gestion de l’information / pipelines / logiciels utilisés tant que la donnée est pertinente et que les logiciels sont accessibles à des utilisateurs lambda pour la gestion reporting / comptabilité / fiscalité. (reporting hebdomadaire ou mensuel).

Votre directeur appuie cependant sur le caractère volatile de ses nouveaux actifs et considère que des outils classiques ne suffiront pas à de bonnes prises de décisions. Il vous demande de penser à un système résilient, écoutant le marché en continu (les cryptos actives ne ferment jamais à l’inverse de la bourse classique) ainsi que d’analyser les tendances des différents acteurs du marché (Gouvernements, institutionnels, influenceurs).



## Objectifs pour la v 0.1

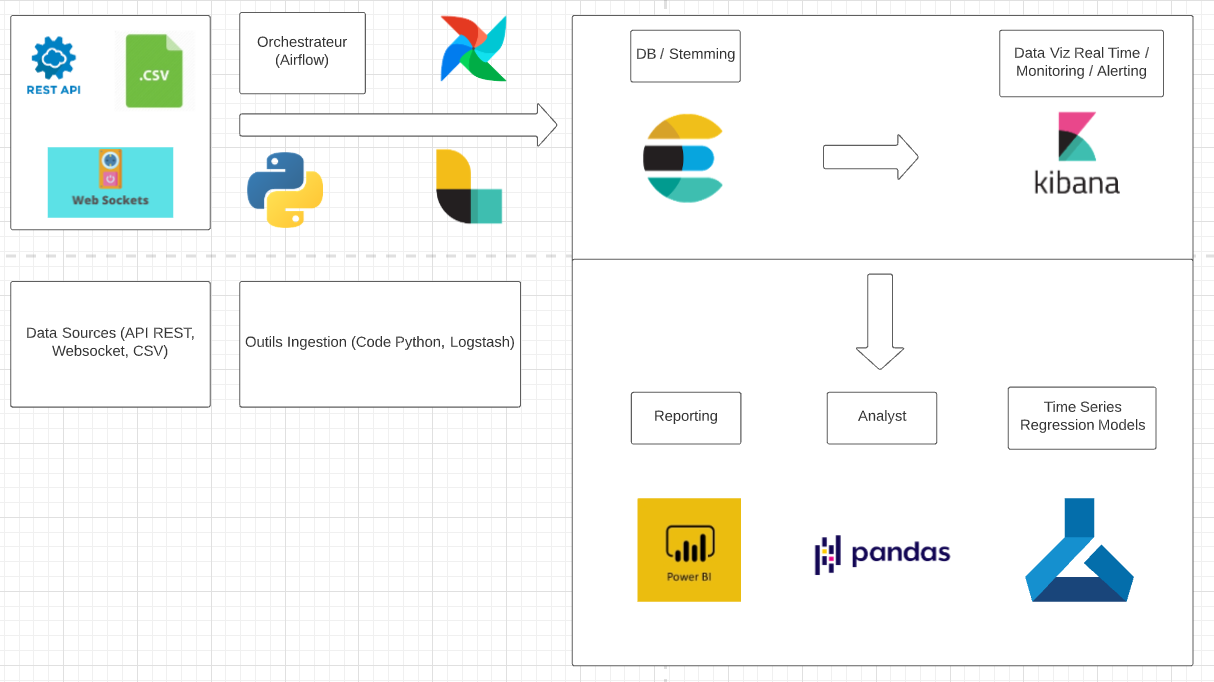
* **Prendre connaissance du fonctionnement du marché dans sa globalité.**
* **Etablir une architecture adaptée permettant l’ingestion en temps réel d’un volume conséquent de données ainsi qu’un logiciel pour des utilisateurs réalisant des traitements sur des données archivées (batch) pour des reportings.**
* **Sensibiliser nos responsables financiers (DSI / Manager) au secteur des crypto-actifs par l’émission d’un rapport de synthèse et d’analyse relatant les différents indicateurs du marché ainsi que l’émission de KPI / points à surveiller.**

## Principaux indicateurs clés

* **Top 10 crypto en capitalisation boursière et volume de transaction de 2017 à 2021. Intégration en interne car gestion de la donnée / traitement / conservation historique.**
* **Prix par unité de 2017 à 2021. Comparaison cours des monnaies. Spread (High / Low à la journée). 2017/10 à 2021/07 : données intégrées via CSV. Chaque monnaie = 1 index.**
* **Détermination et calcul des saisonnalités.**
* **Comparaison Gold / DOW / S&P500.**



## Conceptualisation et Représentation



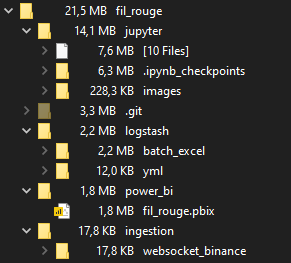
* **Utilisation d’API / de serveurs Websocket et de fichiers CSV pour la récupération de données en temps réel et de données historiques.**
* **Import de la donnée via des scripts python et des fichiers .yml dans Logstash. L’ensemble sera orchestré par Apache Airflow pour relancer les scripts en cas de défaillance de l’hôte ou du serveur.**
* **Ingestion de la donnée dans une base de données (DB) Elastic Search avec Kibana pour la visualisation de la donnée historique et en temps réel.**
* **Envoi sur Power BI et Pandas pour analyse (reporting et machine Learning).**

# Ingestion

## Gestion de projet et travail collaboratif

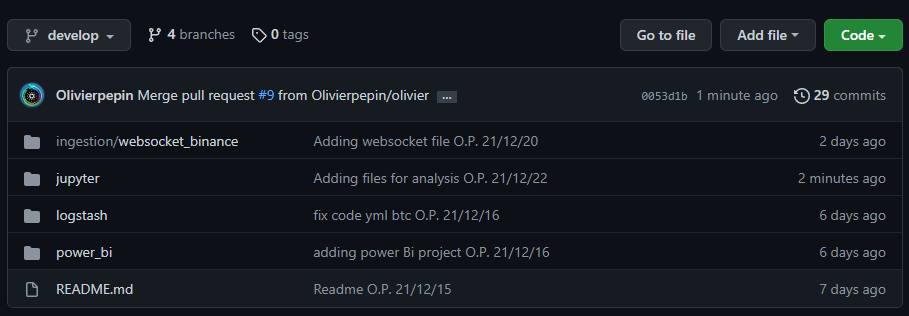
La première journée est dédiée à l’échange sur le secteur des cryptomonnaies ainsi que à la mise en place d’une structure de projet pour le travail collaboratif. Etant deux sur le projet, il est important que le projet soit partagé sans pour autant empêcher l’un des développeurs de travailler avec des données à jour.

Nous avons donc arrêté une structure de projet de cette forme :



* jupyter : pour les analyses liées à Pandas
* logstash : pour l’ingestion des données historiques dans ES
* power bi : pour les fichiers et reportings de power BI
* ingestion : comprend les scripts Python liés aux websockets / API

Pour synchroniser notre travail : nous avons fait le choix d’ouvrir un repository public sur Github. Le repo dispose de quatre branches : deux branches de développement pour chaque développeur, une branche pour merge les features / bug fixes fait sur les branches de devs et enfin la branche Master qui accueillera les versions stables (dans notre cas la v 0.1).

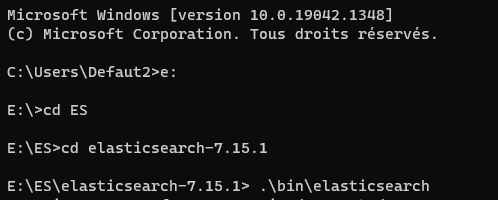


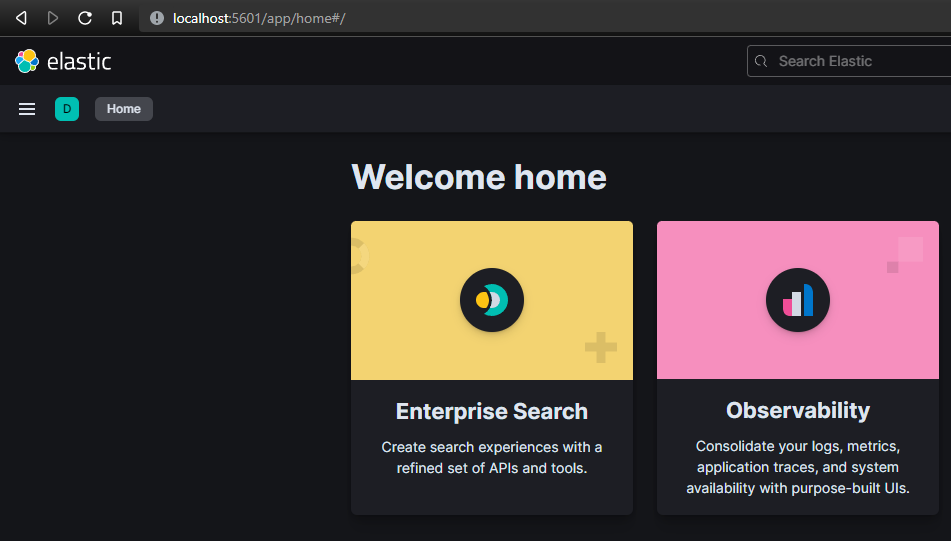
## Installation de la base de données

Nous allons utiliser Elastic Search (ES) comme évoqué au-dessus. Il est particulièrement bien adapté pour traiter de la donnée provenant d’internet (architecture API REST par exemple) ainsi que pour du traitement en temps réel.

Son installation est des plus simple. Il suffit de télécharger ES / Kibana et Logstash sur le site officiel de Elastic (<https://www.elastic.co/fr/>) pour une installation en local ou via des images docker. Dans notre cas : les fichiers sont en local sous Windows.

Il faut ensuite lancer les services qui sont par défaut sur le 9200 pour ES et 5601 pour Kibana.





Par default, ES n’inclut pas de sécurité. Nous avons donc activé la gestion des rôles directement dans son fichier config.

## Conceptualisation des données dans ES

Les données proviennent principalement de deux types de données :

* Fichiers csv / JSON pour les archives (batch)
* Données en temps réel via des Websockets / API

Pour commencer il faut nous constituer une DB dans ES de données historiques. Par souci de simplification nous nous limiterons à 10 tokens avec sa contrepartie en USDT (token / USDT) car le USDT (Tether) est la paire la plus utilisée. Le Tether est la représentation numérique du Dollar Américain.

Listes des 10 tokens :

* ADA / USDT
* BNB / USDT
* BTC / USDT
* DOGE / USDT
* DOT / USDT
* ETH / USDT
* LINK / USDT
* LTC / USDT
* SOL / USDT
* XRP / USDT

Il faut alors définir de quoi seront composés nos documents dans nos index. Pour simplifier le traitement et les requêtes : chaque index s’appellera « {$nom\_token}\_historical »

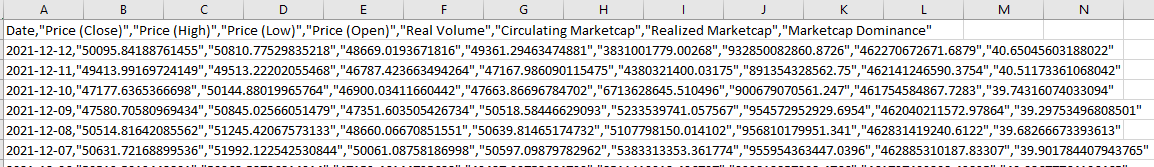
Caractéristique d’un document :

* Date (en jour)
* Price Open (prix du token en début de journée) / float
* Price Close (prix du token en fin de journée) / float
* Price High (prix le plus haut atteint dans la journée) / float
* Price Low (prix le plus bas atteint dans la journée) / float
* Circulating Supply (Nombre de token en circulation) / integer
* Circulating Marketcap (price \* circulating supply) / integer
* Marketcap dominance (% détenu du marché Crypto-monnaie) / float en %
* Volatility (volatilité) / float en indice

Concernant la donnée en temps réel : Nous n’avons pas besoin de maintenir les documents générés à la seconde une fois la journée terminée. Cependant nous souhaitons conserver toutes les données de la journée. Il faudra donc prévoir une requête pour récupérer les données (curl) à 24 heures et les envoyer dans les index historiques.

## Ingestion via Logstash

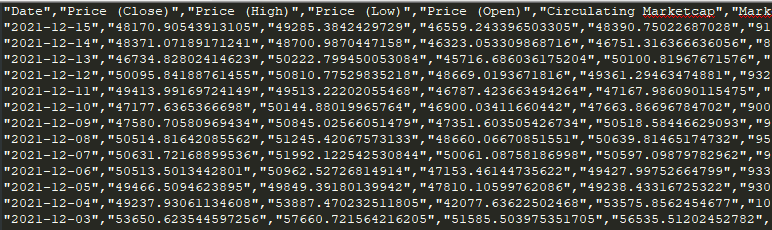
La donnée historique (batch) sera notre base pour commencer les traitements. Il existe différentes méthodes pour récupérer de la donnée historique. La meilleure solution reste la récupération sur des sites spécialisés dans la mise à disposition de données avec service pro / premium. Le site messari.io est un bon exemple. Qui plus est, l’ensemble des informations nous est accessible sans abonnement premium. Une fois les différents champs requêtés, regardons le fichier en sortie.



Que pouvons-nous constater :

* Il manque un champ « Nom »
* Il y a des caractères spéciaux dans les en-têtes de données
* Les chiffres sont au format string avec beaucoup de chiffres après la virgule.
* Il manque des informations sur les derniers jours.

Attention il est préférable de procéder à une vérification des fichiers autrement que par Excel qui modifie l’affichage. Un outil comme NotePad (NotePad++ de préférence) offre un meilleur affichage.



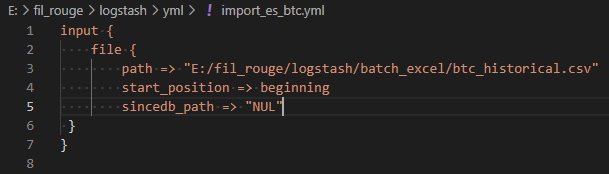
On voit bien que l’intégralité de nos données sont en format String. Dans notre cas, nous souhaitons avoir des dates ainsi que des integers / floats. Il est donc préférable de remplacer l’ensemble des « " » par du vide. Un simple Find & Replace (F&R) suffira. Au vu de notre projet, nous intégrons qu’une dizaine de fichier identique dans leur format. Il n’est donc pas nécessaire de créer un script automatisant le process. Il est plus simple de faire un « F&R all documents » pour ne faire qu’une fois le traitement. Cependant si la tâche était cyclique, une automatisation via un script serait plus pertinente (gain de temps et réduction des erreurs).

Faire une intégration en l’état nous donnerait de la donnée peu exploitable. Il faut donc passer par un nettoyage (cleaning de la donnée). Pour cette étape, nous disposons d’un très bon outil, à savoir Logstash.

Logstash fait partie de la suite ELK et a pour but la collecte, l’ingestion de données structurées et non structurées. Grâce à des filtres et des plugins : il est possible d’effectuer des ingestions simultanées et même de listening sur le fichier en cas de nouvelles données à importer suite à l’ingestion.

Un fichier yml fonctionne avec différents blocs réalisant chacun une tâche. Dans notre exemple, il y a principalement un bloc input pour lire le fichier d’origine, le bloc filter pour le cleaning / rajout de données et le bloc output pour l’ingestion dans ES.

Bloc Input :



Rien de bien compliqué concernant les options. « Path » pour la localisation du ficher, « start\_position » pour la position de la lecture dans le fichier, « sincedb\_path » pour la gestion de la dernière position lue. Dans notre cas nous n’en avons pas besoin d’où le NUL.

Bloc filter :



La partie csv contient :

* Columns : prise en charge des différentes colonnes pour ingestion
* Separator : le séparateur utilisé dans le fichier
* Skip\_header : permet de skip les entêtes du fichier

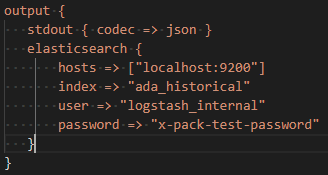
Date permet de faire correspondre la date à un format de date désirée. Dans notre cas, nous n’avons pas besoin d’une précision inférieur au jour. Un simple année-mois-jour suffit. Il est cependant conseillé de faire correspondre les dates au format ISO 8601 soit yyyy-MM-dd (<https://fr.wikipedia.org/wiki/ISO_8601>).

Le plugin mutate permet de réaliser des processus comme de la modification/transformation et suppression. Notre première utilisation sera pour changer le type de nos valeurs. Stockées par défaut en string, nous avons besoin des dates en format date pour nos données en temps réel (de type *time series*) et de integer / floats pour des calculs / analyses.

Par défaut les fichiers ne contiennent pas le nom du token. Par souci de clarté pour les futures analyses : il serait bien que les données soient liées au nom du token et non pas seulement au nom de l’index. Cela sera surtout important pour les query et analyses Pandas. C’est pour cela que *document* contient un champ token correspondant au nom du Token.

ES rajoute des champs pour les versions, le host, l’intégralité du message et le chemin du fichier. Nous n’en avons pas besoin dans notre document. Ils sont donc supprimés par un filter « remove\_field ».

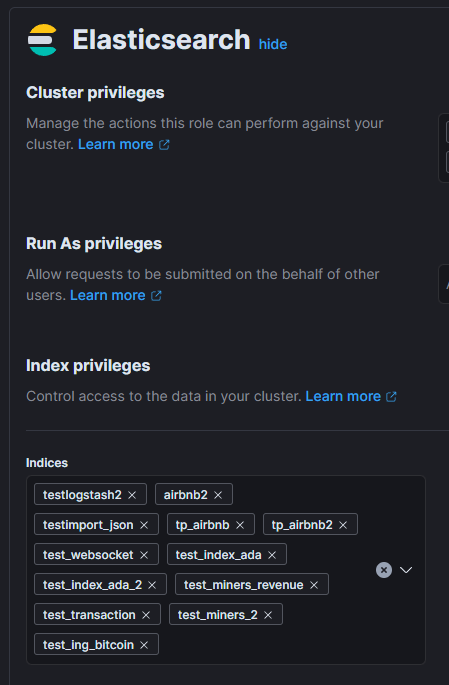
Bloc Output :

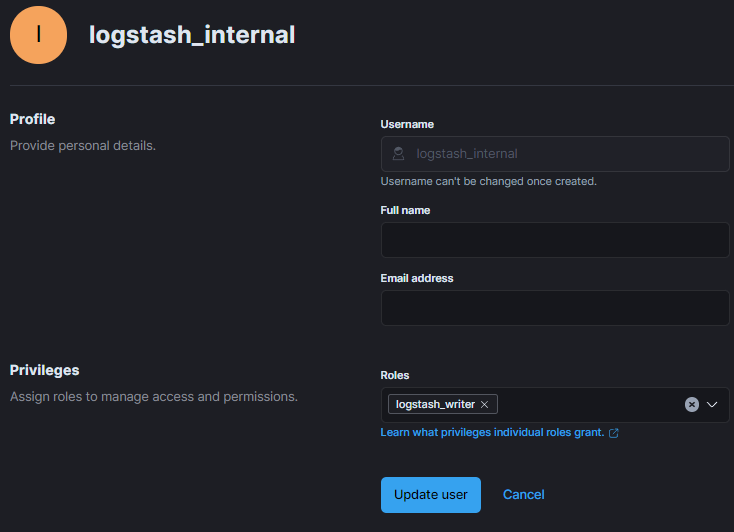


Il contient :

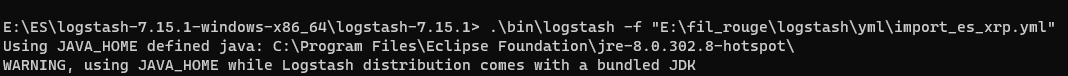
* Stdout : correspond au flux de sortie de l’application dans la console. Pour une meilleure lecture des informations il est sorti en format JSON.
* Hosts : correspond à l’envoi des données pour ES. Actuellement en local (local host ou 127.0.0.1). Les données sont envoyées sur ES qui est par défaut sur le port 9200.
* User / Password correspond au rôle qui peut écrire directement sur ES.

Il y a donc bien une gestion des rôles du côté de ES à faire. Pour cela nous devons créer un user, lui définir des privilèges ainsi que des privilèges sur des index. L’ensemble de cette configuration se fait directement dans le management via Kibana (port 5601).





Une fois les fichiers yml mis en place et les droits attribués : Il suffit de run logstash dans la console. Un « -f » permet de configurer un chemin pour lancer un YML.



Nous devons vérifier sous Kibana si l’ingestion s’est bien déroulée.





Il est important de vérifier dans la console et dans ES que l’ingestion s’est bien déroulée nous évitant de perdre des précieuses minutes à l’avenir. Comme nous avons skip les headers de nos fichiers, le nombre de documents doit être égal au nombre de lignes de nos fichiers -2, car Logstash ne considère pas une ligne sans retour chariot (i.e., il faut une ligne vide à la fin de chaque fichier).

Vérif du csv xrp\_historical via Notepad++ :

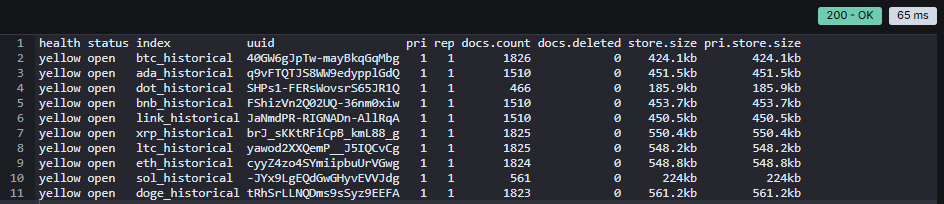




Nous sommes donc bons sur cet index pour l’ingestion et pour la cohérence des données se trouvant dans « source ».

Vérification des index (avec utilisation de Wildcard) :





Point concernant logstash : notre process logstash listen toujours notre fichier xrp\_historical. Cela signifie que en cas de modification et sauvegarde, la nouvelle donnée sera intégrée automatiquement tant que le process continuera à s’exécuter.

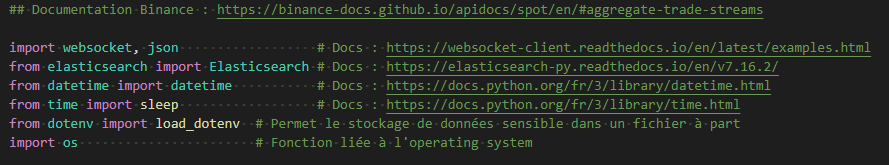
Nous disposons dorénavant de nos données historiques pour notre base de données. Il faudrait dorénavant tester cette donnée via des query / des dashboards Kibana ainsi que des extract pour BI et Pandas. Mais avant cela, l’ensemble de ces fichiers doit être envoyé sur notre Github.

## Ingestion et visualisation des données en temps réel via Python

La notion de temps réel est subjective. De manière générale on considère qu’un système est en temps réel lorsque ce système contrôle un procédé physique à une vitesse adaptée à l’évolution du procédé contrôlé. Dans notre cas : nous avons besoin d’une échelle de temps à la seconde car le secteur des cryptomonnaies peut enregistrer des volumes très importants de transactions par seconde (TPS) pouvant aller jusqu’à 50 000 transactions par seconde sur certaines blockchains.

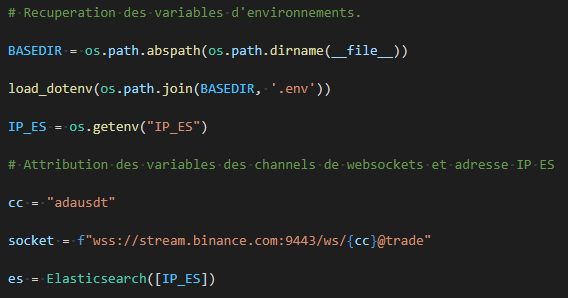
Par chance, les principaux exchangers (Binance, Kraken, Coinbase etc) mettent à notre disposition des websockets afin de pouvoir récupérer ses données en temps réel avec juste un peu de code (et de lecture de documentation).

Ci-dessous un script permettant la récupération du payload d’un websocket de Binance :

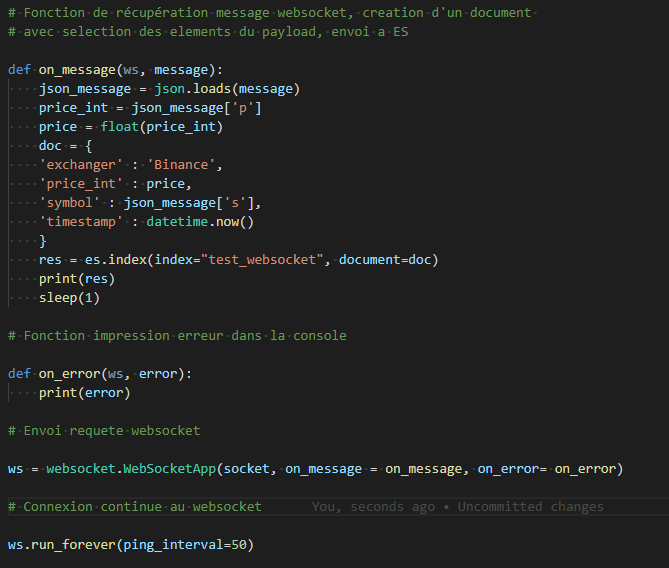


Nous importons en premier lieu nos dépendances afin, d’un côté, interroger le websocket et de l’autre l’envoi de données à ES.

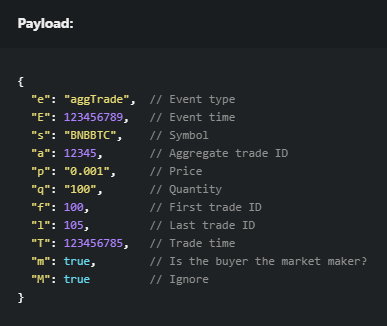
DotENV nous permet de stocker nos données sensibles dans des fichiers à part.



Cette partie correspond à la récupération des variables d’environnements ainsi que à l’attribution des différentes variables et constantes. Notre script sera quasi identique pour chaque token (sauf pour les variables). Dans notre cas : il est préférable de créer une multitude de scripts pour ne pas perdre l’intégralité de notre données en temps réel (temps réel = pas de batch).



La dernière partie du script récupère le payload. Il est donc possible de créer un document sous forme de dictionnaire selon les clés du payloads, à savoir :

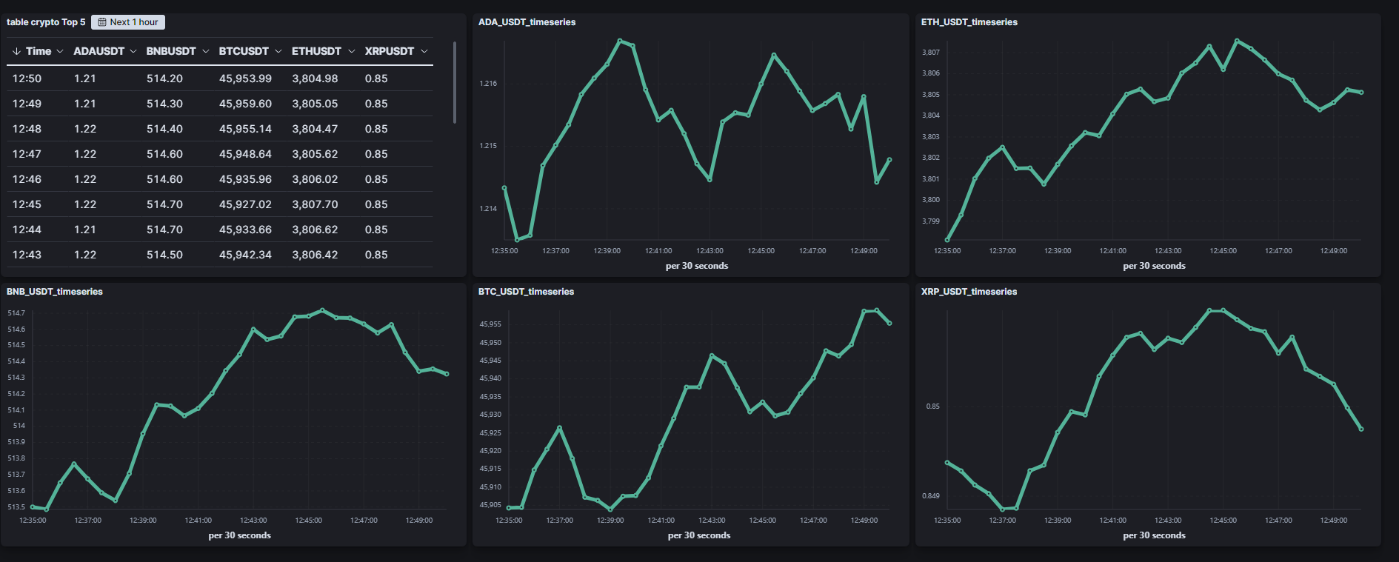


Une fois notre dictionnaire réalisé, il suffit de l’envoyer dans le bon index de ES (qui nous renvoie un accusé). La fonction « run\_forever » se charge de maintenir la connexion au websocket avec un ping de réponse au serveur pour ne pas fermer la connexion. En cas de deconnexion, la fonction « on\_error » nous affichera la cause dans la console.

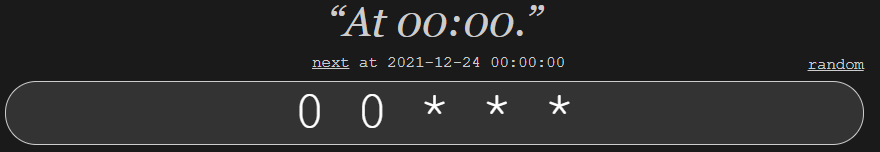
Si nous souhaitons dorénavent visualiser nos données via Kibana, il faut créer un *index pattern*. Grâce à lui, nous pourrons produire des dashboards.



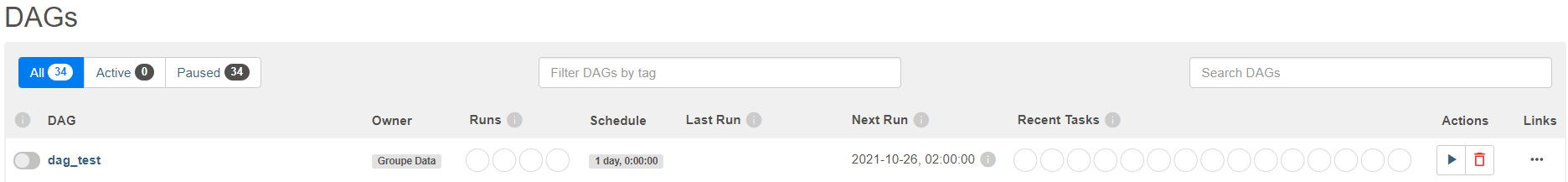
Dashboard prix crypto temps réel avec un rafraichissement de 30 secondes :



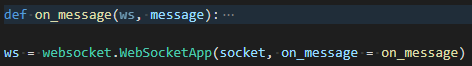
Concernant les requêtes API : c’est le même principe que les scripts Python sauf que le script commence par des cURL pointant sur le bon endpoint permettant la récupération des informations, la transformation en documents et l’envoi dans un index historique tout cela géré par un programme executant du Cron toutes les 24 heures à minuit :



Afin d’obtenir des données en temps réel (time series), nous avons élaboré un script python qui envoie des données sur Elastic Search à partir d’une API Websocket. La Websocket est plus conciliante qu’une API de type REST et ne bloque pas si trop de requêtes sont effectuées dans un délai court puisqu’il s’agit d’une écoute de la donnée, fonctionnant avec un système de pings au serveur.

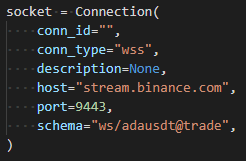
Néanmoins, le script retourne une erreur au bout de 15 à 18 minutes, ce qui interrompt son exécution et provoque une perte de précieuses données. Afin de répondre à cette difficulté, l’utilisation d’un logiciel d’ingestion semble adaptée. Le langage utilisé dans le cadre du projet étant python, Airflow semble le plus pertinent, et propose un scheduling qui correspond à nos besoins.

Le script doit donc être adapté pour être utilisé via un dag. Problème : il repose déjà sur une définition de fonction avec un callback de ladite fonction fait via la librairie Websocket. Cette structure est difficilement interprétée par Airflow qui retourne une erreur « missing arguments », ne comprenant pas le paramètre « on\_message » :





De plus, le lien hypertexte de la Websocket n’est pas compris par Airflow et demande une décomposition sous forme de classe afin de créer un Connector, l’une des spécificités de Airflow.



Ne parvenant pas à résoudre l’erreur « missing arguments », utiliser un logiciel d’ingestion pouvant accepter le script tel qu’écrit initialement a été notre idée. Apache NiFi en était capable, c’est pourquoi il a été choisi. Malheureusement, NiFi n’a pas reconnu la librairie Websocket et n’a pas pu aboutir à un lancement du script.

Malgré plusieurs jours de recherche, les compétences nécessaires pour comprendre les erreurs associées aux Websockets ont freiné notre progression sur les deux solutions d’ingestion que nous avons proposées, il a donc fallu oublier l’automatisation de la relance des scripts pour la première version du projet et la reporter à plus tard, avec des recherches plus approfondies.

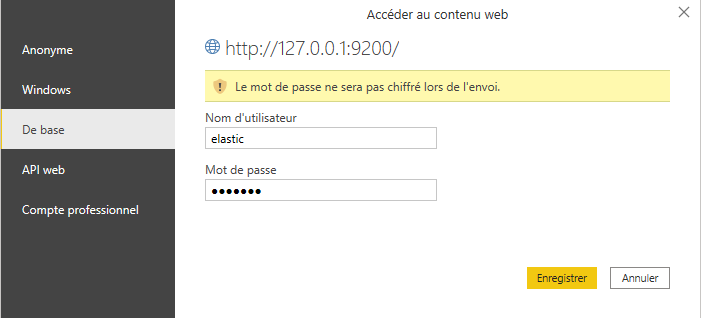
## Tests de connectivité Power BI et Jupyter.

Une fois nos flux de données branchés à notre DB : nous devons brancher nos logiciels.

Nous allons utiliser la force de ES pour récupérer nos informations et réaliser un gain de temps précieux grâce aux requêtage HTTP. Pour Power BI : après l’installation du logiciel en local et l’ouverture d’un projet, il nous suffit d’authentifier un utilisateur et de rentrer une query similaire à un endpoint terminant par le nom de l’index avec les paramètres supplémentaires que l’on désire.

Power query se chargera de la transformation de la query et de la mise en forme avant intégration de la donnée grâce à son propre langage : le langage M.

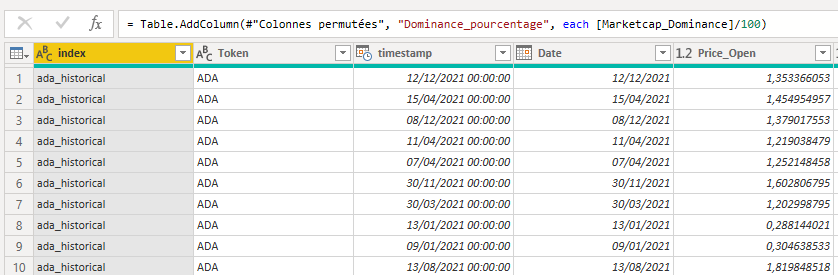
Authentification :



Query et ingestion via M :



Le résultat nous donnera une table similaire à celle de Access.



Les requêtes sont enregistrées directement dans Power BI, permettant de raffraichir les tables quand nous le souhaitons afin d’intégrer de nouvelles données.

Nous avons dorénavent branché Power BI. Il ne nous reste plus que notre notebook à connecter.

Pour utiliser Jupyter (et plus précisement Jupyter lab), nous avons besoin dans un premier temps besoin de monter le service. Tapons jupyter lab dans un terminal et le service se monte par défaut sur le port 8888.

La connexion à ES est quasi identique que sur Power BI. Il nous suffit de faire une requête ES grâce à la librairie Elasticsearch (déjà utilisé dans les scripts Python). Nous pouvons dès lors envoyer cette donnée dans un objet ou un dataframe pour commencer nos travaux d’analyse.

Test connextion ES / Jupyter :



C’est ainsi que se conlut cette partie dédiée à l’ingestion. L’ensemble des branchements sont opérationnels et l’objectif fixé pour la v 0.1 est réalisé. Pour les versions ultérieures, il faudra en priorité améliorer la stabilité des websockets et trouver un remplaçant à notre orchestrateur pour la suivi des scripts et l’automatisation des tâches.

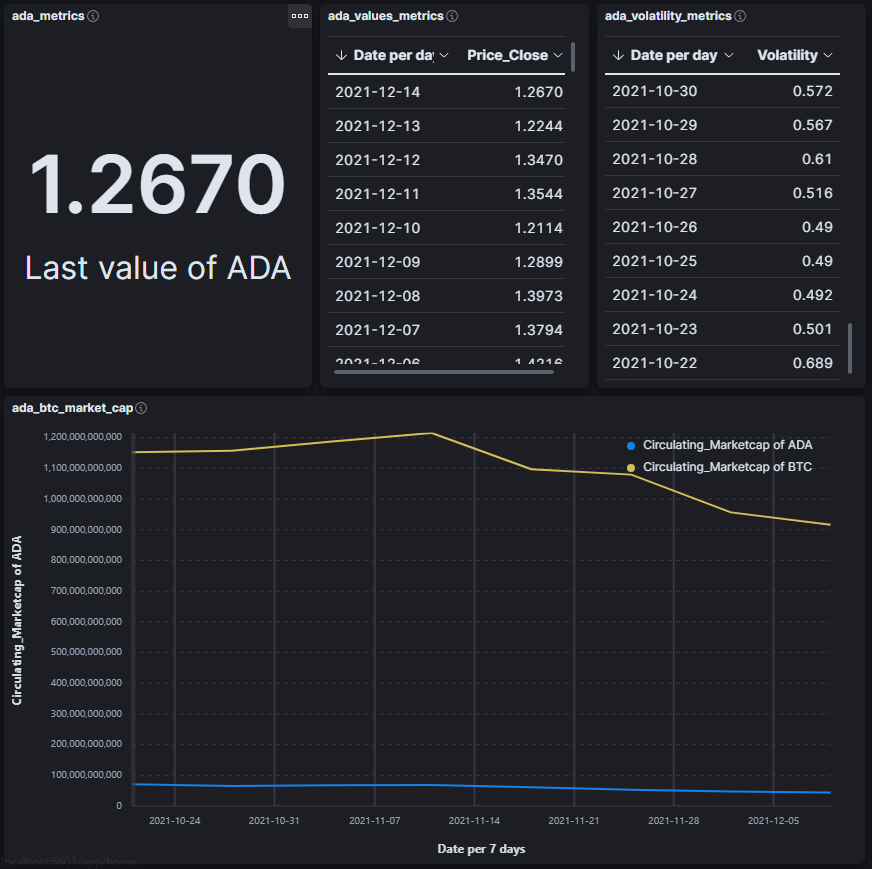
Nous pouvons dorénavant commencer notre phase d’analyse et de reporting.

# Analyses et visualisation

## Visualisation via Kibana

Grâce à Kibana nous pouvons réaliser un grand nombre de graphiques pour nos visualisations de données historiques ou en temps réel. Après un petit temps d’adaptation, le logiciel permet la réalisation de graphes dynamiques complexes pouvant être intégrés dans un dashboard. Chaque visualisation est indépendante et permet d’être *Drag & Drop* d’une vue à une autre.

Exemple de dashboard :

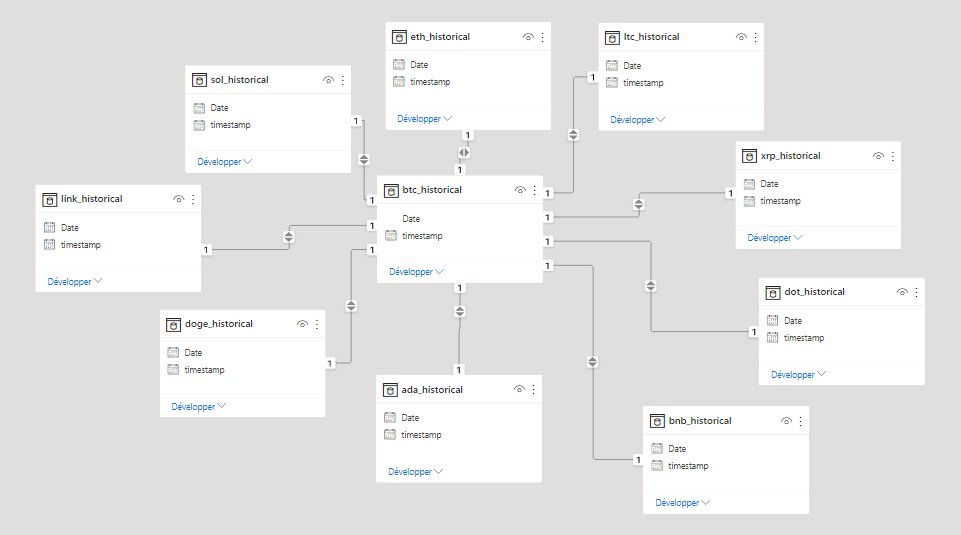


Bien que complet pour du suivi en temps réel : Kibana n’offre pas autant d’outils que des logiciels comme Power BI en terme de représentation avancé ou encore comme Pandas avec Python. Il restant cependant amplement suffisant pour réaliser des présentations via ses canvas et pour du reporting.

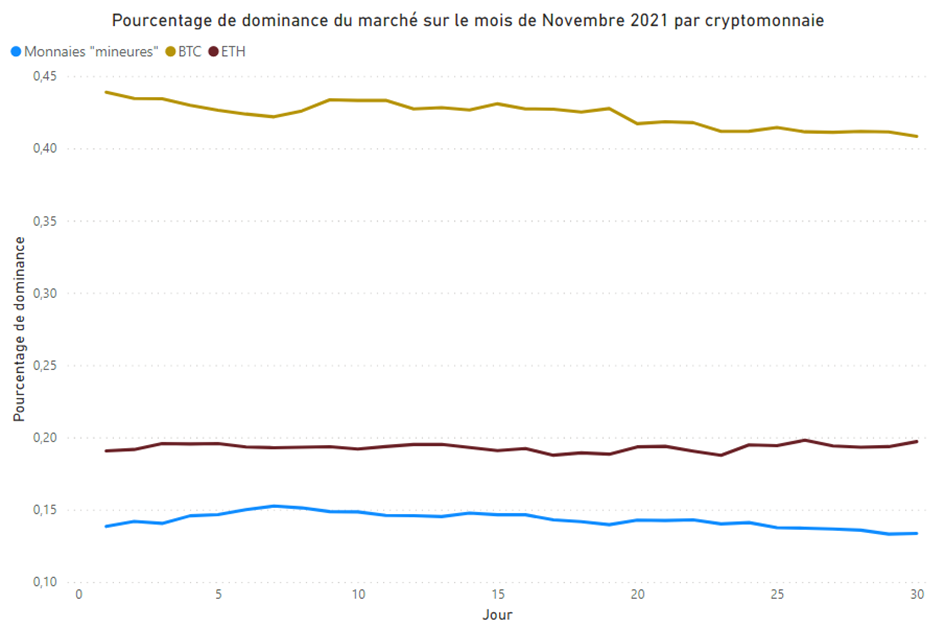
## Visualisation et analyses via Power BI

Avant de démarrer nos visualisations, il nous faudra lier nos tables. La donnée provenant d’une source non-structurée, nous avons besoin de travailler avec un modèle de type relationnel étant donné le fonctionnement de Power BI.

Relations Power BI :



L’analyse des données historiques sur power bi a commencé par un état des lieux sur la domination du marché par les 10 plus importantes. Le choix de la période s’est porté sur le mois de Novembre 2021 afin d’avoir un mois complet d’analyse permettant de vérifier les fluctuations existantes et d’avoir des données récentes, plus parlantes dans le cas de la problématique du projet.

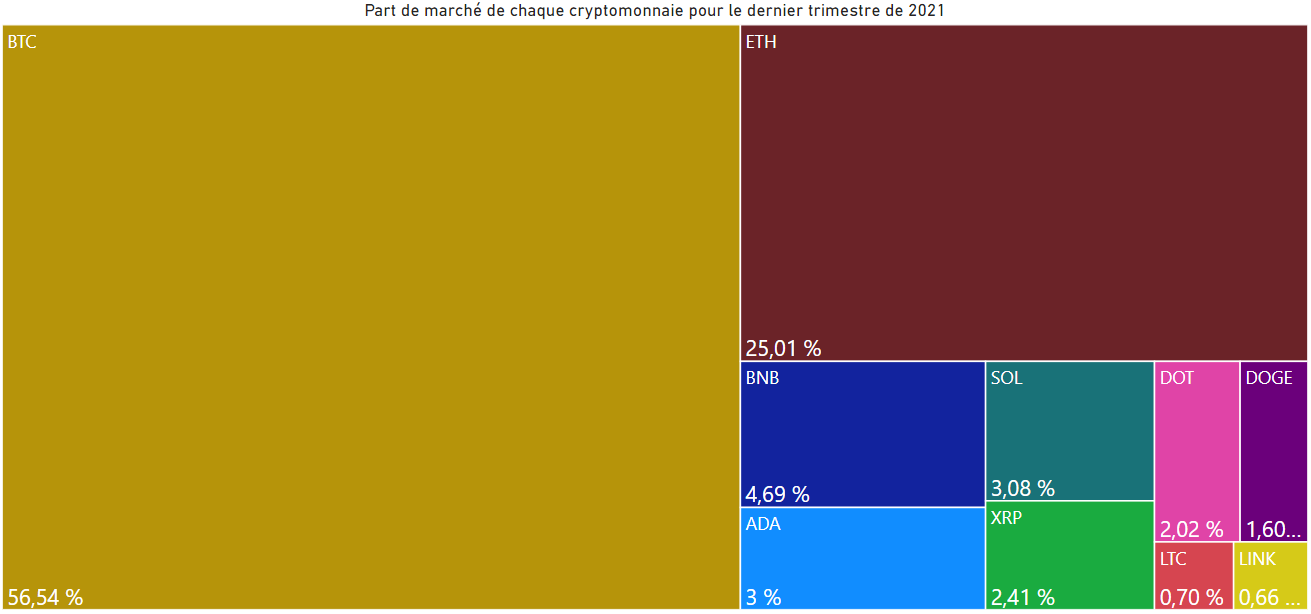


Trois niveaux peuvent être distingués, un premier pour le bitcoin qui représente une part considérable du marché, en dessous l’ethereum qui se distingue des autres monnaies inférieures et les cryptomonnaies « faibles » pour lesquelles il était nécessaire d’effectuer un regroupement afin de pouvoir les observer.

Tandis que le bitcoin perd un peu de dominance au cours du mois, les autres monnaies sont stables. Il est possible que l’émergence d’une ou de plusieurs cryptomonnaies puisse expliquer cette baisse pour le bitcoin, toutefois ces données historiques ne sont pas suffisantes pour l’affirmer.

En dehors de la domination, la capitalisation boursière de chaque token a été étudiée. Elle est obtenue par le produit de la somme des tokens en circulation multiplié par le prix unitaire.

La capitalisation boursière des crypto-monnaie étant quelque chose influencée par des paramètres mondiaux (économie, COVID, inflation etc…) nous avons pris une périodicité au trimestre (dernier trimestre 2021) pour lisser ces paramètres.

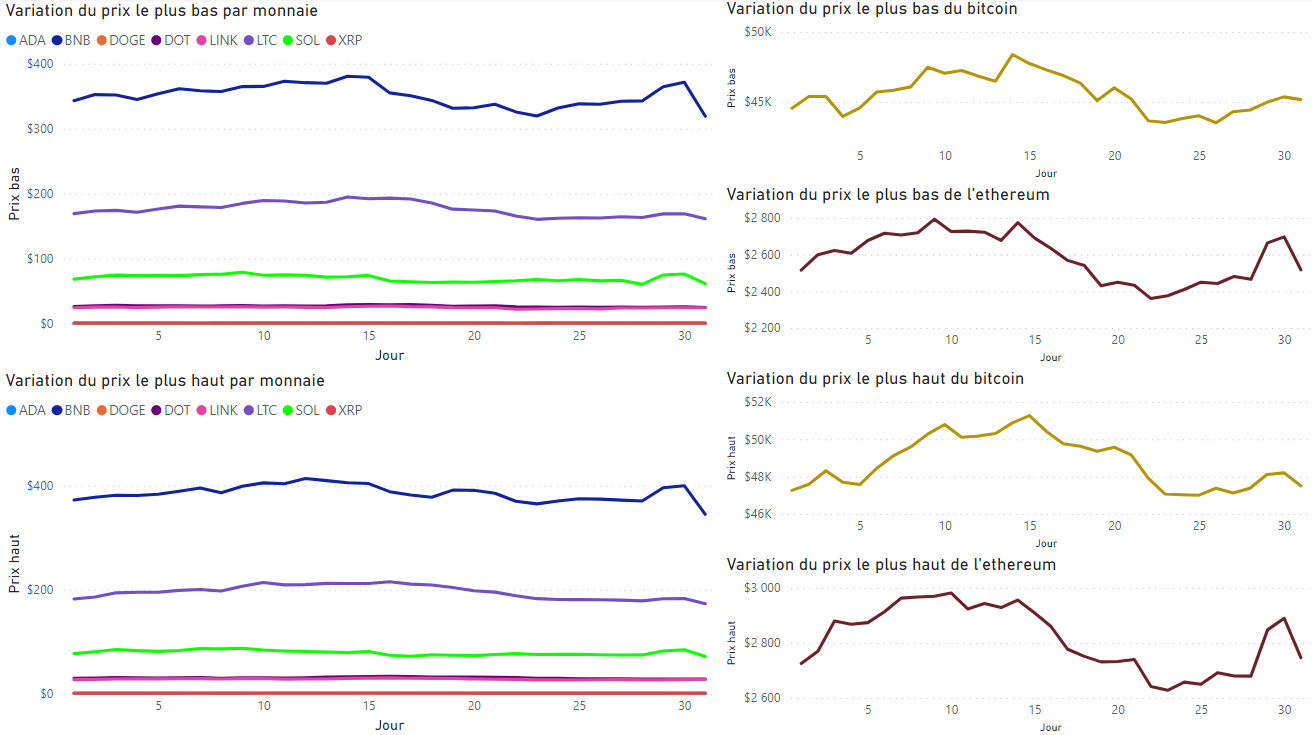


Le même rapport entre les monnaies demeure sur cette treemap. Seulement, le bitcoin est davantage présent, tout comme l’ethereum. Cela s’explique par leur valeur importante face aux autres cryptomonnaies, ainsi que par la prise en compte des monnaies du top 10 seulement. Le graphe précédent prenait en compte l’ensemble des tokens existants.

La question qui suivit fut de savoir quel jour était le plus propice à un investissement et quel jour il faudrait choisir pour obtenir des gains.

En se limitant à une analyse historique, il est plus intéressant d’investir à partir du 20ème jour du mois et avant le 25ème. Le 15ème jour du mois suivant, il est possible de récupérer un bénéfice.

En s’intéressant à l’année 2021 ou pour préparer l’année 2022, le 22 ou le 23 de chaque mois semble le plus intéressant pour déposer des fonds, tandis que du 10ème au 15ème jour affichent les prix les plus élevés pour un retrait.

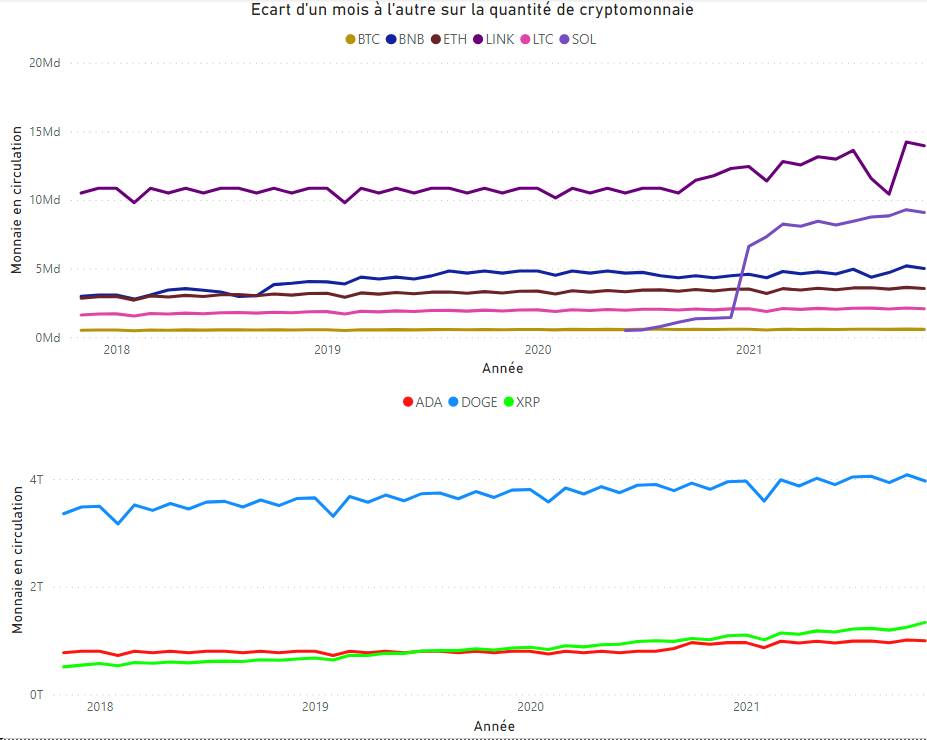


Un phénomène supplémentaire est observable : les jours qui enregistrent les prix les plus haut ont des variations des prix les plus bas les plus élevés du mois.

De plus, chaque 31 enregistre des valeurs plus basses. C’est une erreur d’interprétation de power BI qui ramène à 31 jours tous les mois et qui compte donc des 0 tous les mois ne comportant pas 31 jours.

Cette analyse présente deux limites. Premièrement, le cours des cryptomonnaies est constant et ne peut pas être limité à des analyses globales. Le cours connait des variations différentes d’un mois à l’autre. Deuxièmement, choisir un jour pour investir n’est pas suffisant pour déterminer le meilleur moment pour investir. En effet, le prix varie tout au long de la journée et seule une analyse à la seconde offrirait des insights suffisants pour prédire un moment clé pour investir. La présente analyse avertit surtout sur les jours de vigilance lorsqu’il y a une volonté d’investissement.

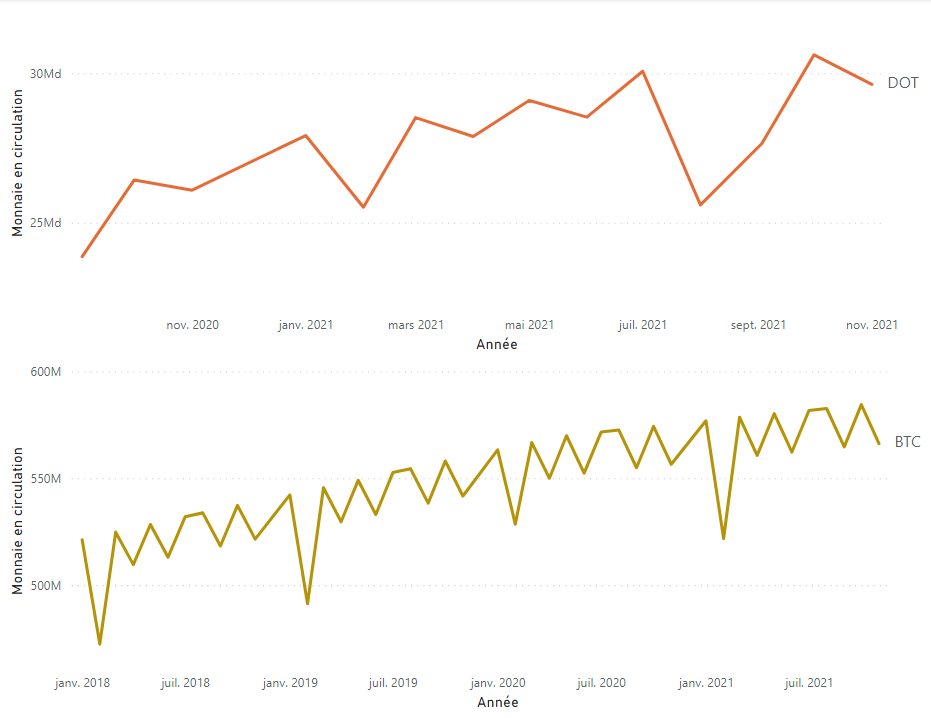
Enfin, nous nous sommes intéressés à la quantité de monnaie en circulation sur toute la période disponible. Les tokens sont brûlés pour certaines monnaies tandis que d’autres acceptent de produire sans définir de limite. Chaque monnaie du top 10 mériterait une analyse à part entière juste sur ce point, c’est pourquoi nous avons privilégié une présentation plus générale qui soulèvera ultérieurement des points à approfondir.



Tout d’abord, il est clair que la quantité de monnaie en circulation est maintenue à un niveau le plus constant possible. En cas de baisse comme pour le DOGE, de la monnaie est produite au cours du mois suivant pour rétablir le niveau. La chute du LINK observé cet été 2021 a aussi été rattrapée en un mois.

Pour l’émergence de tokens tels que le SOL, on voit une production progressive. Une levée du plafond du nombre de SOL en circulation peut expliquer son explosion début 2021.

Portons à présent le focus sur le DOT, absent des graphiques du dessus pour des raisons de lisibilité et le BTC.



A son échelle, le DOT fluctue fortement sur toute la période, sans proposer de pattern réel. À l’inverse, le bitcoin montre une saisonnalité, augmentant progressivement tout en cherchant la stabilité maximale.

Le constat général est que chaque cryptomonnaie produit plus de monnaie en circulation au fil du temps. Pour certaines, les écarts sont marqués et importants tandis que d’autres fluctuent plus légèrement et se stabilisent aussi plus rapidement.

## Visualisation et analyses via Jupyter et Pandas

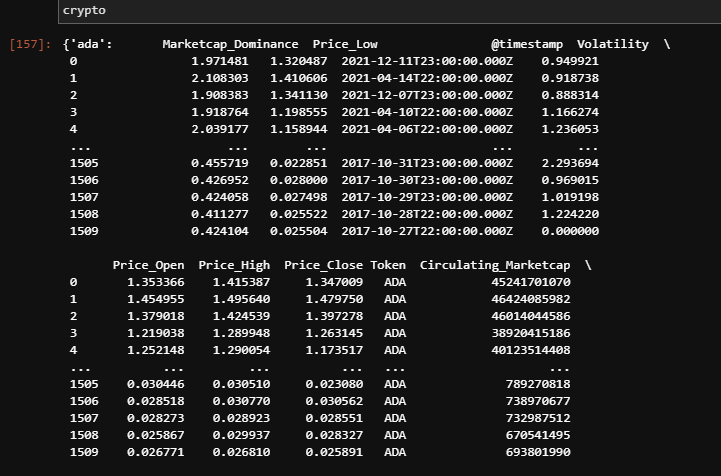
Afin de répondre à notre problématique, l’analyse tournera autour de trois axes principaux :

* Analyse de la Capitalisation boursière dit Marketcap
* Analyse des prix
* Analyse de la volatilité

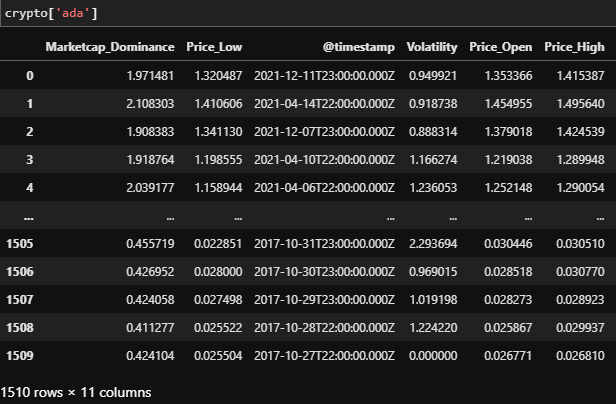
### Analyse de la Capitalisation boursière

Pour procéder à l’analyse, l’ensemble des index ont été mis dans un seul dictionnaire afin de récupérer / interroger notre donnée facilement. Les index étant imbriqués : il suffit de spécifier le nom de l’index sur notre dictionnaire pour accéder aux données. Pour récupérer une donnée spécifique de chaque index, il suffira de rajouter des boucles « for » / « while » ou des fonctions lambdas.

Dictionnaire crypto :

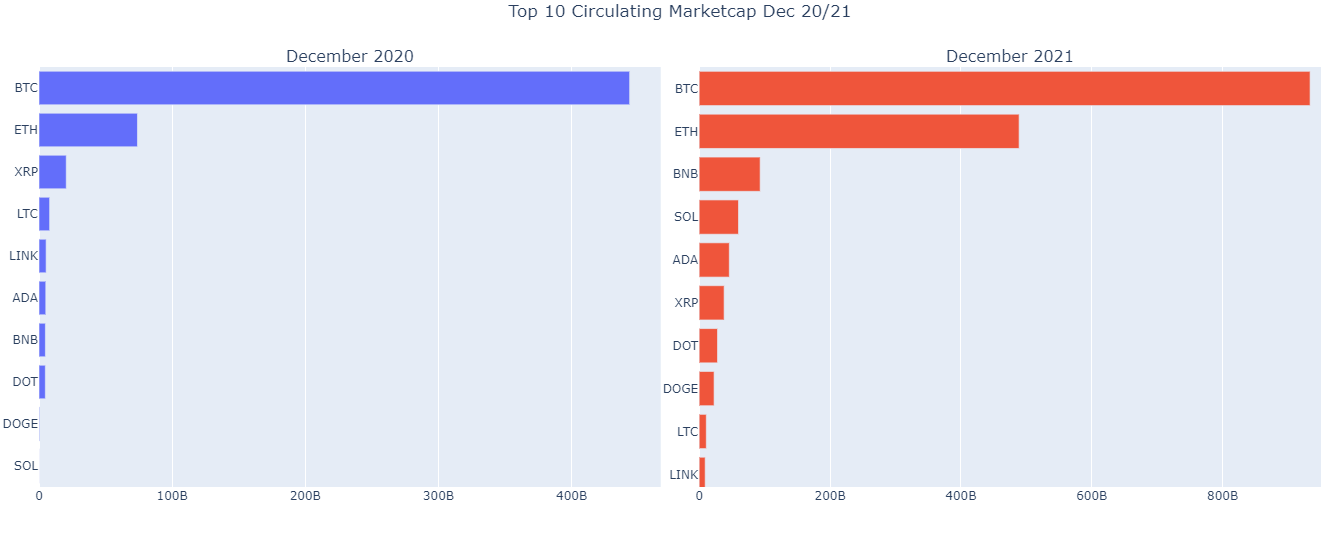


Accès à un Dataframe d’un index :



Nous allons dans un premier temps analyser les dix valeurs les plus importantes du marché. Cela nous servira de proxy. Les 10 premières valeurs représent plus de 70% du poids du marché, le reste des 30% étant réparti sur plus de 10 000 tokens. Il serait bien trop complexe d’intégrer cette masse non pertinente (token trop récent, pas d’audit dessus, etc.).

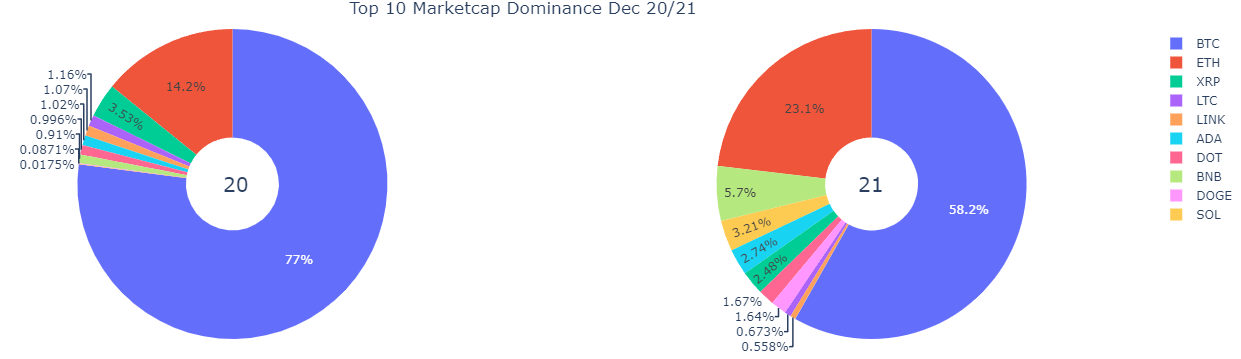
Une représentation en bar chart est intéressant pour ce genre de données. Pour montrer l’evolution, une comparaison avec N-1 (au 15/12/21) est une bonne solution.



Principaux résultats :

* Le marché est passé de 600 milliards à plus de 1 700 milliars de $ (USD) en 1 an.
* Le Bitcoin (BTC) et l’Ethereum (ETH) restent majoritaires en grande partie.
* Chaque valeur a au mininum doublé son market cap (5x ADA, 6x ETH,) en 1 an.
* Bien que encore dominant, le BTC perd des parts de marché.

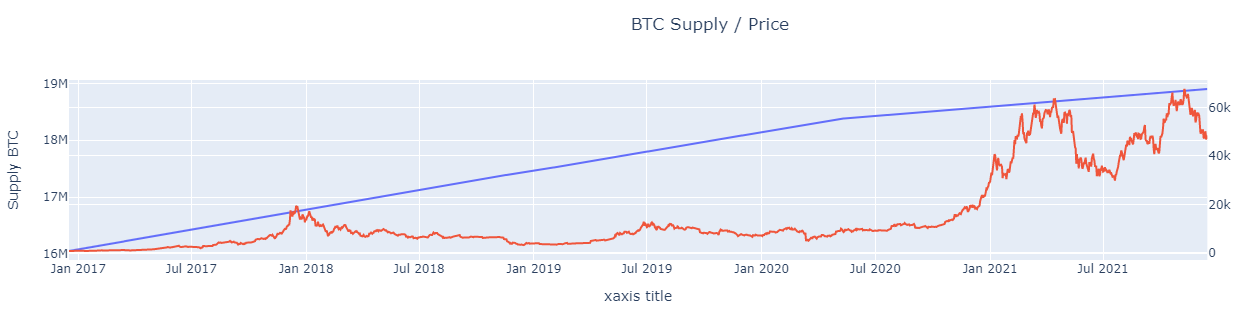
Pour continuer, nous allons analyser la dominance du marché par rapport à N / N-1 :



Principaux résultats :

* Le BTC cède plus de 13% de ses parts, principalement à l’ETH et au BNB.
* Le poids des trois premières monnaies est passé de 95 % à 87 %
* Il faut dorénavent 0,55 % du marché pour être dans le Top 10 contre 0,01 % en N-1

Cette analyse démontre que l’étalon, le BTC, bien que majoritaire n’est plus le seul élément de ce marché. Cela s’explique principalement par l’émergence d’une nouvelle génération de tokens intégrant des contrats intelligents, dits smart contracts, qui sont difficilement réalisables sur le BTC. L’injection massive de monnaie fiduciaire (USD principalement) a aussi créé une diversification dans les portefeuilles, recherchant en priorité des actifs plus risqués à plus forte valeur ajoutée que le BTC. Un autre élément ayant fait monter le prix est le supply du BTC qui évolue moins vite que la demande (cf. graphique ci-dessous).



Maintenant que nous avons analysé les principaux acteurs de ce marché, attardons-nous plus sur le prix.

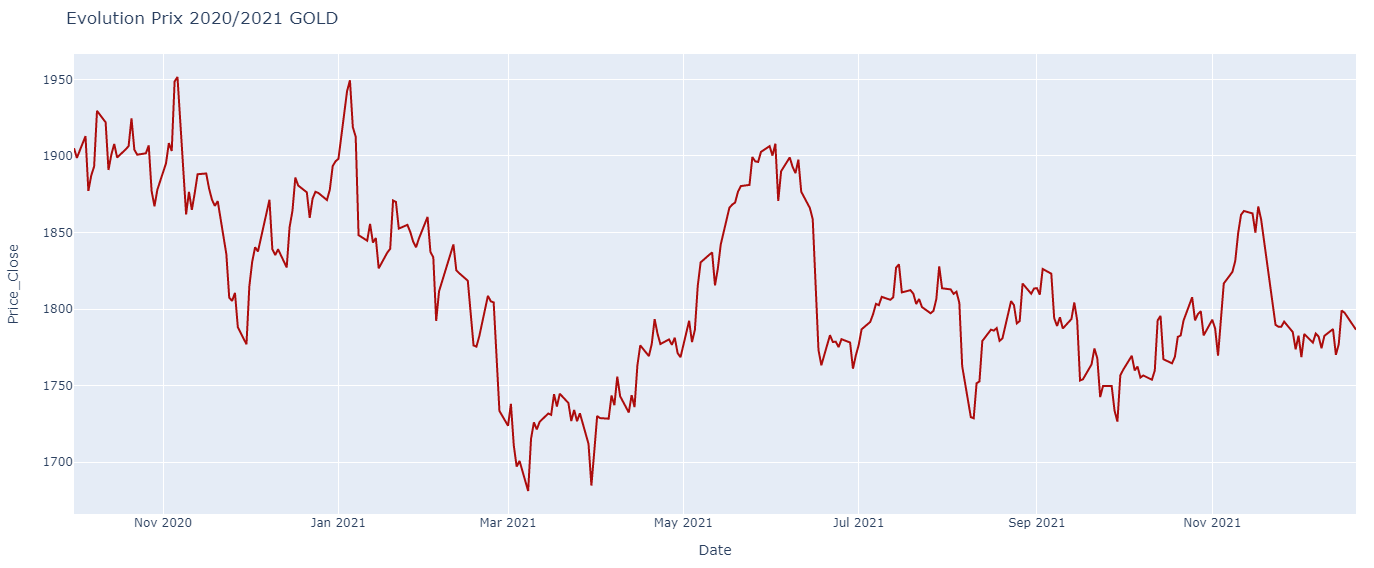
### Analyse des prix

Pour bien se rendre compte de l’évolution des prix, rien de plus parlant qu’un line chart.



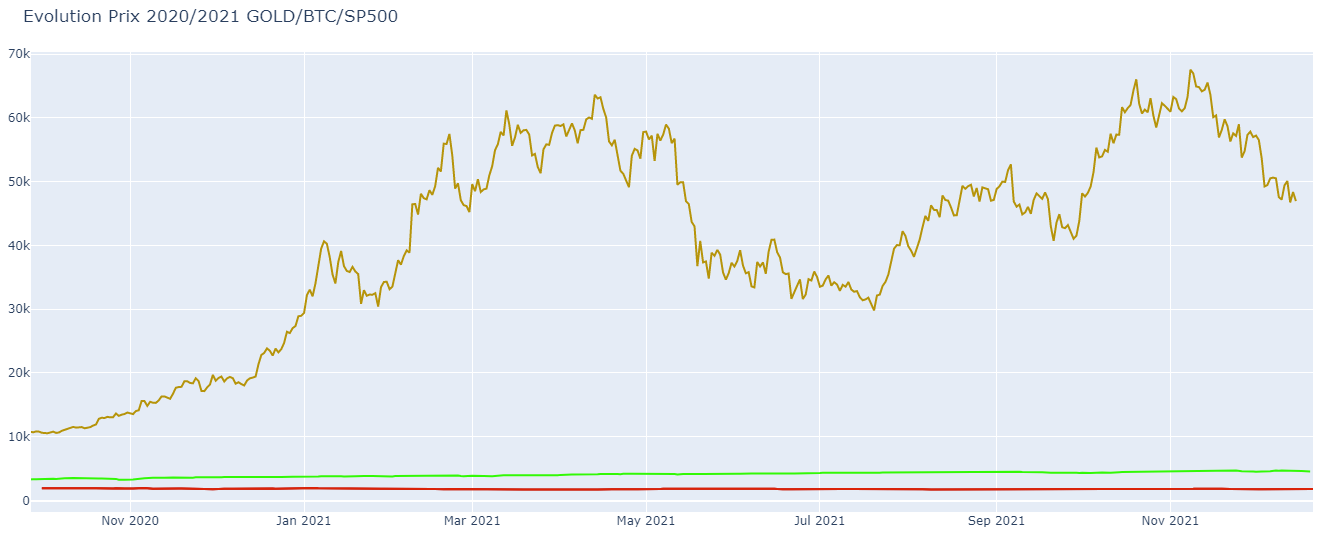
Ce graphique représente l’évolution du BTC depuis son redémarrage au 4ème trimestre 2020 à maintenant. Son prix étant passé de 10k $ à 47k $ en passant par des pics à 67k $. La tendance est donc clairement à la hausse même si nous constatons une grande volatilité dans le prix, ce qui est compréhensible au vu de l’explosion du prix ainsi que l’évolution de la masse monétaire injectée dans le marché. Dans l’ensemble, la plupart des autres crypto-monnaies suivent cette tendance. Il y a une corrélation positive entre l’évolution du prix du BTC et les autres cryptos.

Afin de rattacher ses chiffres à des sphères plus classiques : il serait intéressant de voir l’évolution des valeurs refuges en comparaison. Et qui parle de valeur refuge, parle de l’or !



On constate bien que l’or n’a pas bougé durant cette période avec une légère tendance à la baisse. Pas de saisonnalité non plus. Il est probable qu’une partie significative des capitaux de l’or aient été drainés par le Bitcoin, expliquant cette légère tendance à la baisse.

Pour vraiment enfoncer le clou sur les valeurs classiques : comparons le prix du Bitcoin, l’or et le S&P500 (indice boursier sur 500 grandes sociétés américaines).

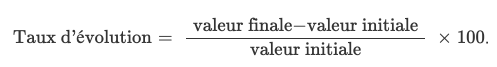


Il n’est plus à démontrer l’explosivité du Bitcoin par rapport aux valeurs classiques et de refuge sur cette période. Il ne faut cependant pas oublier que l’or est une valeur stable depuis des millénaires et que le S&P500 représente les actifs de sociétés cotées en bourse, ce qui lui donne un caractère légitime à l’instar du Bitcoin et son côté purement spéculatif. Le S&P500 a bien augmenté en 1 an (plus de 24%) mais n’est que peu perceptible sur cette échelle au vu de l’envolée du prix du BTC.

Il est donc indéniable que la rentabilité à court terme du Bitcoin explose les produits classiques. Cependant le Bitcoin reste un actif jeune soumis à une grande incertitude sur son devenir. Le principal reproche qui lui est fait est sa volatilité. C’est pour cela que nous terminerons cette analyse sur la volatilité du secteur des crypto-monnaies.

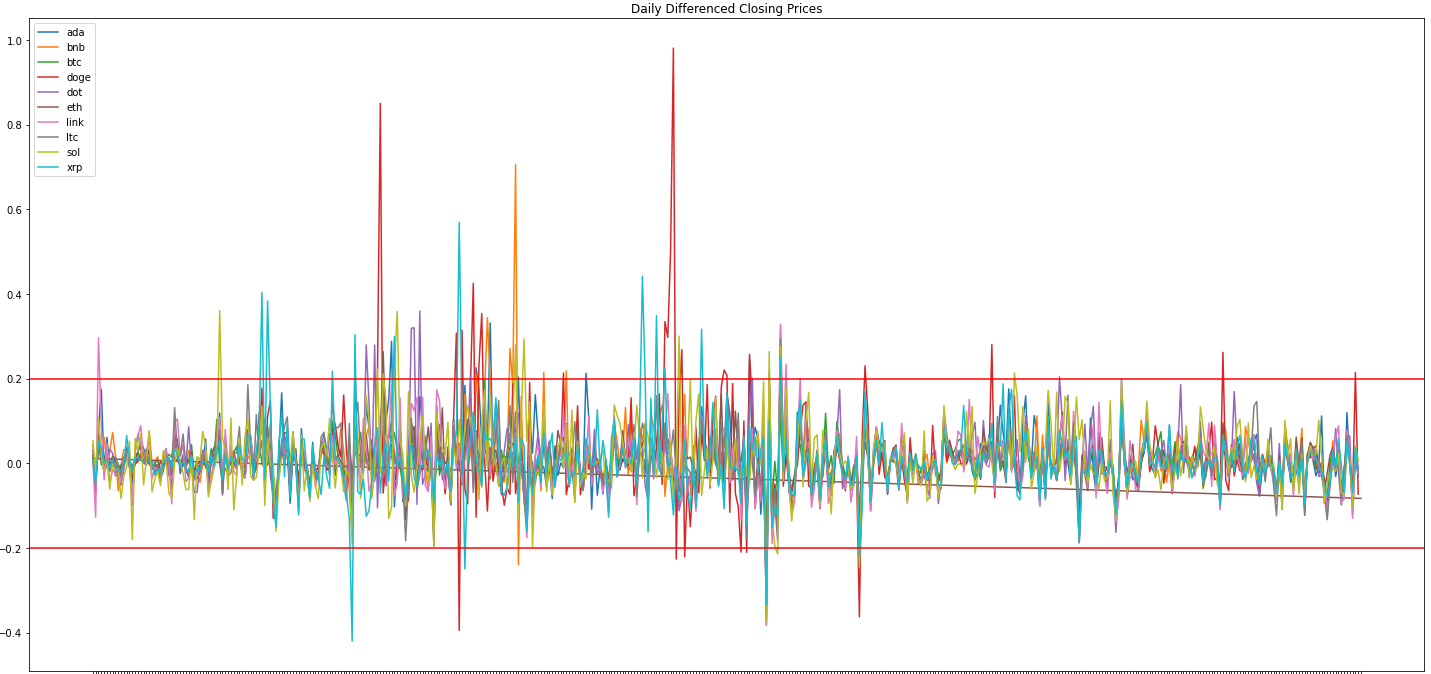
### Analyse de la volatilité

Afin de calculer sans avoir des problèmes d’échelle sur les valeurs (un BTC vaut 47k $, un ETH 4k $...), il est préférable d’effectuer cette analyse en pourcentage d’évolution. Le plus simple est d’appliquer la formule de la variation à savoir :



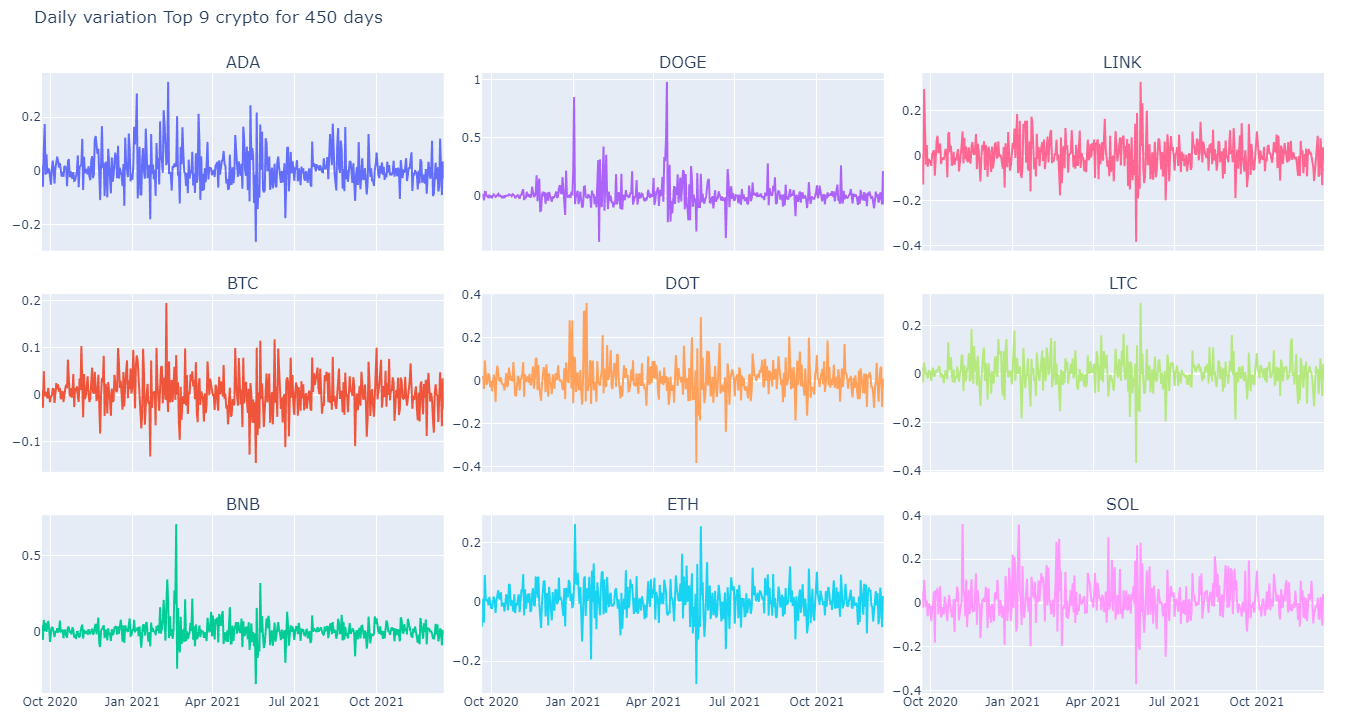
Il suffit alors de créer une colonne de variation dans chacun de nos index qui prendra la clôture d’aujourd’hui moins la clôture de hier.

Ci-joint la représentation pour l’ensemble des dix cryptos d’Octobre 2021 à Décembre 2021 :



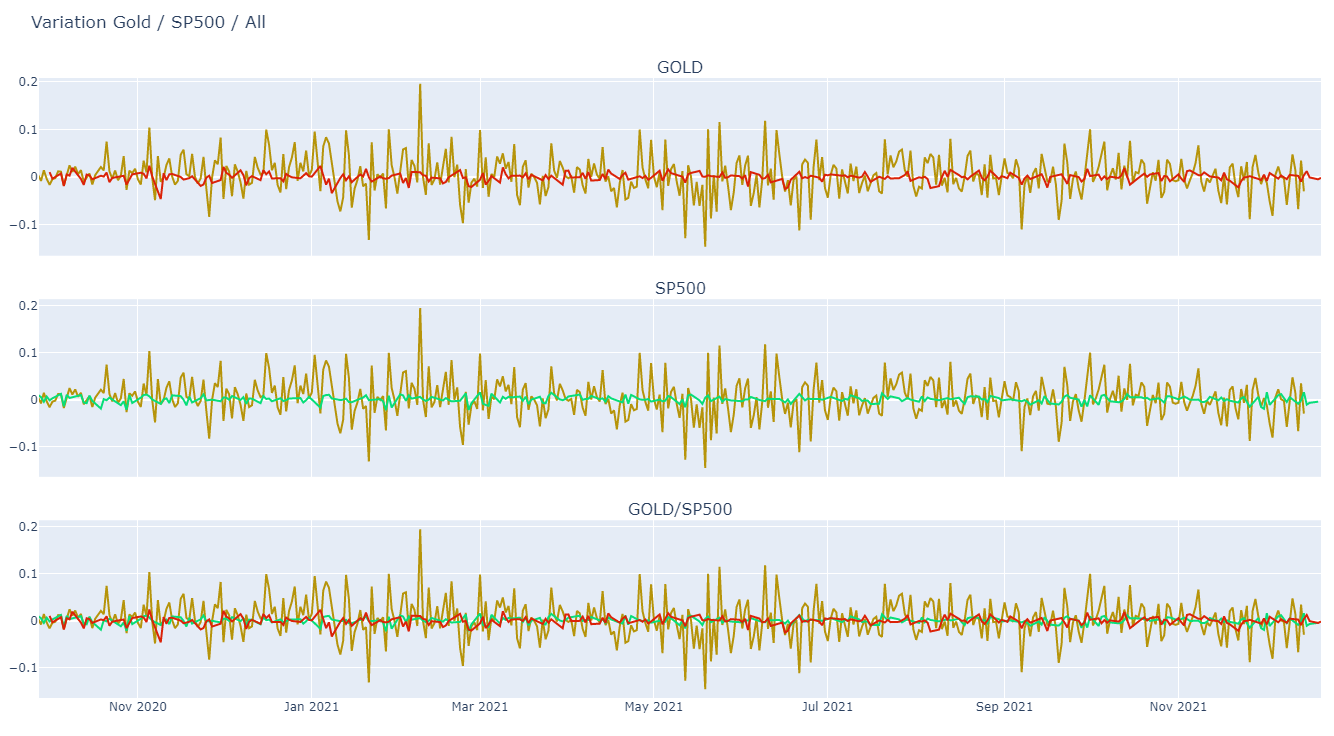
Par souci de lecture, une donnée du DOGE réalisant une variation de + de 4 a été ôtée du graphique (une analyse à part de ce token sur sa volatilité serait intéressante). Dans l’ensemble, on constante beaucoup de bruit avec des variations pouvant aller de presque 1 jusqu’à -0,4 dans la même journée. En moyenne la variation journalière est contenue dans un intervalle de -0,2 à 0,2.

Les crypto-monnaies ne démordent donc pas sur leur caractère volatile. Cependant chaque cryptomonnaie n’est pas soumise à la même volatilité. Plus le marketcap est élevé, plus la volatilité est faible mais le gain sera moindre. Nous pouvons analyser ses volatilités en les compant en subplots pour une meilleure visibilité.



Sans surprise : le BTC a le moins de volatilité suivi par l’ETH et l’ADA. La volatilité du BNB est élevé mais principalement en positif. Nous constatons un pic de volatilité négatif commune en date du 19 mai 2021 correspondant à l’annonce de bannissement des crypto-monnaies en Chine.

Maintenant que nous connaissons les volatilités des crypto-monnaies : nous pouvons les comparer aux valeurs classiques.



Sans surprise : l’or et le S&P500 n’ont que très peu de volatilité (volatilité inclus entre -0.05% et 0,05%). Le BTC est donc en moyenne quatre fois plus volatile que les actifs classiques.

### Conclusion de l’analyse

Cette analyse nous a permis de comprendre les fondamentaux du marché des crypto-monnaies. Le marché est actuellement en oligarchie avec un Top 10 des tokens qui détiennent à eux seuls plus de 70 % de la capitalisation boursière dont le Bitcoin qui en détient encore plus de 40 % à lui seul. Bien que le marché ait plus que doublé en un an, on constate une diminution de la dominance du Bitcoin. Cela s’explique par une diversification des portefeuilles avec une recherche de tokens à plus forte valeur ajoutée.

L’intégralité des tokens ont, au minimum, doublé leur valeur en un an et certains ont réalisé des performances inconcevables sur des actifs en multipliant par 10 leur prix. Il est intéressant de noter que le prix de l’or est plutôt dans une tendance baissière bien que, théoriquement, étant une valeur refuge stable très utilisée lors de crise et d’instabilité économique. Il est probable qu’une partie des jeunes investisseurs (25 – 35 ans) aient préféré investir dans les crypto-monnaies plutôt que dans l’or. Le S&P500 est lui aussi en hausse mais rien de comparable à l’explosion des crypto-monnaies. Il faut cependant attirer l’attention sur le caractère temporaire de ces plus-values exceptionnelles. Au fil du temps, les marketcap augmenteront, le risque diminuera et la rentabilité aussi.

Il reste la problématique de la volatilité. Oui à l’heure actuelle, les crypto-monnaies reste des actifs très volatiles allant de 4 fois la volatilité du S&P 500 à plus de 80 fois pour certaines monnaies. La tendance sur un an reste haussière et la forte volatilité est corrélée à l’envolée des prix.

Au vu de l’ensemble de ces éléments, investir un faible pourcentage de notre trésorerie dans cet actif semble intéressant. Bien que volatile, les rentabilités actuelles sont largement supérieures à celle de produits classiques et ne sont pas soumis à imposition tant que les gains restent en crypto-monnaie.

Une stratégie en DCA (Dollar Cost Averaging) semble le moins risqué et le plus profitable. Une DCA consiste à investir de l’argent à des intervalles réguliers et programmés à l’avance sans tenir compte du prix actuel. Avec une DCA nous émettons l’hypothèse que la tendance reste haussière et qu’importe le prix à un moment T, le prix ne fera que monter au fil du temps (avec des périodes de pump et de dump).

Une analyse de prévision (forecasting) déterminant les points d’entrée les plus pertinents en fonction du jour de la semaine, des semaines de début et fin de mois ainsi que les mois est en cours de réalisation. Cette prévision sera livrée dans la version v 0.2 du projet suivi de l’analyse en LSTM (Long short-term memory) pour la version v 0.3.

En l’état je conseillerais d’investir un montant avec une diversification sur les actifs. De par son caractère dominant, il est préférable d’injecter au moins 2/3 de la somme dans le Bitcoin et ensuite diversifié le dernier tiers dans divers tokens à forte valeur ajoutée avec un projet et une équipe solide.

# Conclusion générale

## Retour sur le projet

De manière générale, le projet s’est bien déroulé et l’objectif de la mise en place d’une infrastructure pour des crypto-monnaies avec des analyses est accomplie. La première semaine fut dédiée à l’analyse du secteur, la mise en place du travail collaboratif et l’ingestion de données.

Les délais ont été respectés dans l’ensemble à l’exception de la mise en place d’un orchestrateur pour les scripts Python. Le principal problème est la gestion des scripts incluant des websockets. Airflow ou NiFi n’apprécie pas ce genre de code, principalement dans la gestion des DAG (Graphe Orienté Acyclique). Après réflexion, nous n’avons pas besoin de DAG dans notre cas actuel car nous souhaitons seulement une relance automatique en cas de plantage d’un script. Pour les prochaines versions : il faudra chercher du côté d’un script Python qui supervise des scripts en continu et les relance en cas de défaillance pour la donnée en temps réel. Le reste des tâches s’est déroulé dans les temps. L’ingestion des volumes de *stable coins* a été cependant déplacé à la v 0.2 du projet.

La deuxième semaine fut consacrée aux visualisations et aux analyses (sur Power BI et Jupyter). L’utilisation de données non structurées sur Power BI était intéressante mais la création de mesures demandait de recréer une mesure par table. Pour créer des dashboards, il serait intéressant de pouvoir automatiser cela ou le produire sous une autre forme. Au-delà de ça, les graphiques disponibles ont permis d’exprimer la donnée autrement qu’avec le temps, donnée pourtant centrale des time series.

La gestion de plusieurs index dans un seul dictionnaire fut déroutante les premières heures mais vraiment agréable pour la gestion de la donnée une fois le temps d’adaptation passé. L’analyse reste généraliste sur la v 0.1 mais importante pour la compréhension globale et afin de revoir les fondamentaux de Pandas et des plots avec Matplotlib et Plotly. Le modèle entrainé grâce à une ARIMA (Autoregressive Integrated Moving Average) ne colle pas parfaitement pour les prévisions sur les transactions par seconde et fera l’objet d’une amélioration pour une intégration dans la v 0.2.

Il ne reste donc plus qu’à modifier certaines fautes et d’échanger sur le merge de la branch de développement pour valider notre version stable v 0.1.

## Améliorations à venir pour la v 0.2 et ultérieurs

* Ajout de Stable Coin (USDT / USDC)
* Amélioration de la stabilité des Websockets / MAJ des scripts de query API
* Gestion des index ES via Index Lifecycle Management (ILM)
* Gestion des replicas ES
* Approfondir les analyses Pandas et le forecasting
* Utilisation de query de type POST via ES
* Utilisation d’un modèle Long Short Term Memory (pour la v 0.3)