Optimisation de stratégie de portefeuille diversifié

Antoine Bigeard, Etienne Cornu, Tina Aubrun, Yoann Launay, Maxime Seince, Murat Can Acun

I. Intro

On se propose ici de former une stratégie de formation de portefeuille optimisé sur des actifs du *SP500* américain.

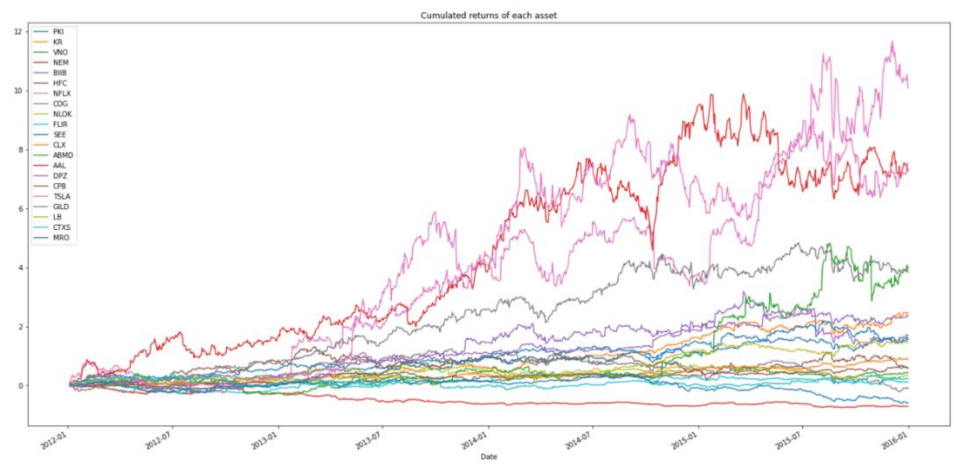
II. Données utilisées

Les données historiques quotidiennes sont issues de finance.yahoo.fr et importées grâce au module python yfinance.

Nous avons fait le choix de travailler sur un marché unique, le *SP500,* marché en *USD*. Dans ce marché nous avons sélectionné entre 10 et 25 actifs de manière non-arbitraire, toutefois indépendamment de leurs performances passées. Dans le cadre d'un portfolio de diversification, il s'agissait en effet de faciliter la décorrélation entre les actifs. Les algorithmes de moindre corrélation sont parfois complexes et c'est pourquoi nous avons choisi d'opter pour un algorithme simple et de notre initiative (voir fonction *suggested_assets*):

- *Prendre un actif initial aléatoirement*
- *Sélectionner chaque nouvel actif en minimisant la somme des valeurs absolues des corrélations (norme 1)*

On en déduit alors les 21 actifs suivants dont les retours cumulés sont représentés ci-dessous sur la période étudiée :



On voit que nous sommes a priori sur une période intéressante pour beaucoup d'actifs.

III. Optimiseur

L'ensemble des options disponibles (et modulables) dans notre optimiseur de portfolio (voir notre module *tools.*py) est résumé dans cette formulation du problème d'optimisation :

$$\text{Minimize : } \overset{(3)}{} \quad \overset{(2)}{w^T\mu_f} + \lambda \frac{w\Sigma}{\sqrt{w^T V w}} - \overset{(4)}{c\left|\left|w - w_0\right|\right|_p} + \overset{(5)}{f_{RSI1}(w,\mu)}$$

$$\text{for } w \in [-1,1] \qquad (1)$$

$$\text{st } \textstyle\sum_i w_i = 1 \qquad (6)$$

$$\text{and } \left|\left|w - w_0\right|\right|_q \leq \tau \qquad (7)$$

$$\text{and } w^T\beta(\mu) = 0 \qquad (8)$$

$$\text{and } w \in I_{RSI2}(\mu) \qquad (9)$$

avec comme données : μ les returns quotidiens sur la période historique d'entrainement, μ_f les returns (début-fin) de la période d'entraînement et Σ et V les écarts-type et la matrice de covariance de cette même période.

Voici quelques éléments de compréhension de ces options qui seront combinées dans la suite :

(1) On maximise un portefeuille normalisé long-short, long ou short selon les cas d'étude. La variable est un vecteur avec autant de composantes que d'actifs. (*long = True/False, short = True/False*).

(2) On maximise en premier lieu le return espéré en prenant le meilleur portfolio sur la période d'entraînement. (*max_return = True*)

(3) On minimise le risque en maximisant le ratio de diversification. Le choix de λ permet de définir la pondération de ce ratio par rapport à la fonction objectif principale (voir (2)). Dans le cas de notre étude, nous mettrons un facteur 10 pour privilégier son optimisation devant le return dans le trade-off. (*diversification = True, risk_parameter = 10)*

(4) Le coût de transaction (*transaction_cost = [c, p]*) est limitant et on peut choisir de minimiser le re-balancement du portefeuille (w_0 étant le portefeuille précédent ou initial ou de référence) en norme *p* en pondérant par *c,* la proportion de frais de transactions associée. Souvent, la convergence est meilleure pour *p = 2* (convexité).

(5) Pour la partie Technical Analysis, nous avons fait le choix d'implémenter le moment du Relative Strength Index (RSI) qui existe sous plusieurs formes. Ici nous avons fait le choix de l'inclure dans la fonction objectif comme suit : le RSI est calculé sur les x derniers jours (voir la fonction *RelativeStrengthIndex*) et lorsque RSI<=30 on encourage linéairement le long et respectivement le short lorsque RSI>=70.

(6) On normalise le portefeuille pour plus de généralité et pour avoir un problème borné.

(7) On peut également limiter le re-balancement en majorant le turnover en norme *q* par rapport au portefeuille précédent. On prendra par défaut la norme *q = 2* (convexité).

(8) On peut également choisir de construire un portefeuille décorrélé du marché SP500 en annulant son β (somme pondérée des β de chaque actif par rapport au *SP500* sur la période d'entraînement). (voir fonction *Betas)*

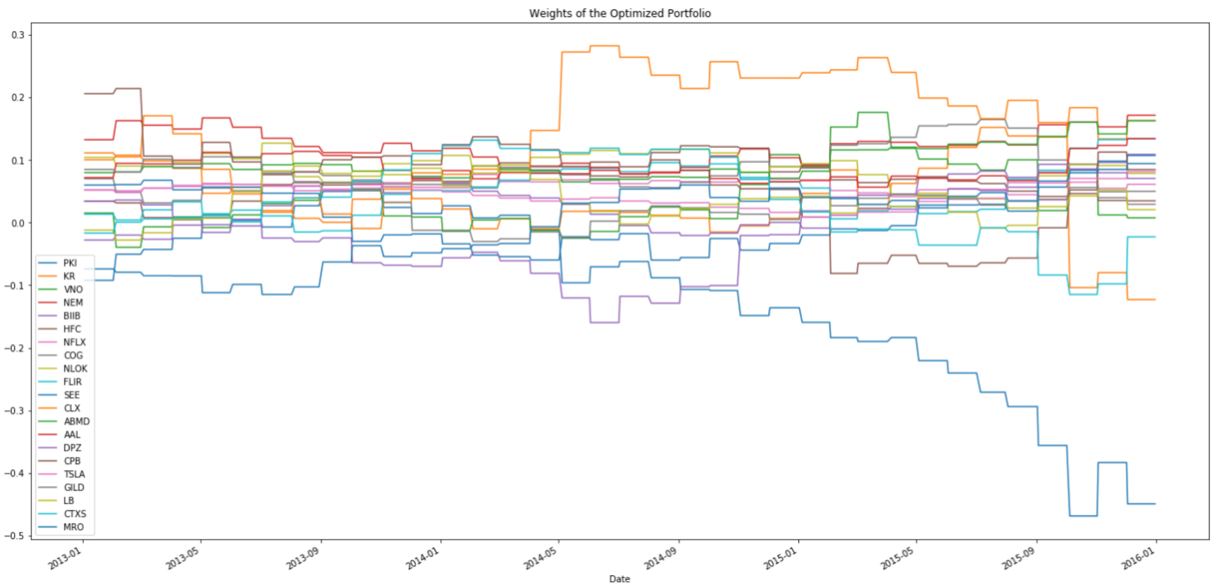
(9) On propose également une autre façon de prendre en compte le moment de RSI de façon plus abrupte et dans les contraintes : lorsque RSI<=30 l'asset est pris long et respectivement il est pris short lorsque RSI>=70.

Tous les portefeuilles obtenus ont été optimisés grâce au module scipy et notamment avec le solver 'SLSQP'.

IV. Rolling Window

Quand nous utilisons un optimiseur sur notre portefeuille, celui-ci considère les données des différents actifs sur une certaine période (1 mois, 6 mois, 1 an, 5 ans…) pour pondérer ces actifs en respectant l'objectif et les contraintes de l'optimiseur. C'est donc une optimisation qui est faite à un instant précis mais dont les poids seront conservés dans le temps malgré l'évolution des différents actifs. Le portefeuille n'est donc alors plus optimal. Ici, nous cherchons à traiter ce problème en actualisant de façon régulière la valeur des poids des différents actifs dans le portefeuille, de manière à garantir, ou du moins à s'approcher, de son optimalité.

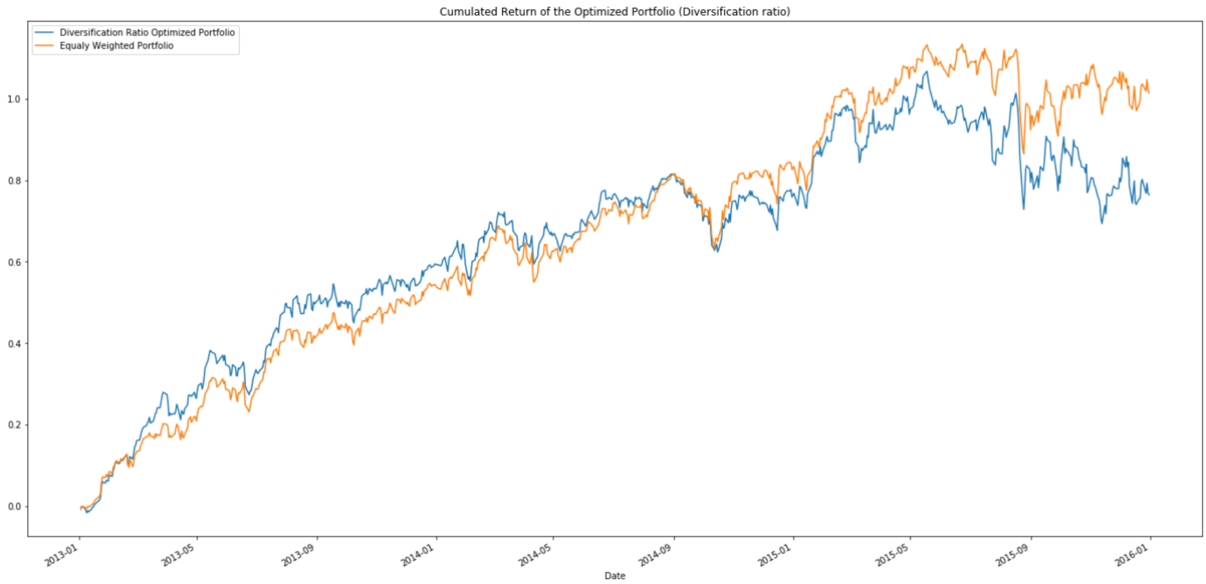
La solution mise en place, dite de *rolling window*, va actualiser les poids à une fréquence fixe. Dans notre cas, cette fréquence est mensuelle, mais il est possible de l'allonger si la réallocation des actifs est plus complexe, 4 fois par an par exemple, ou de la raccourcir pour améliorer l'optimalité. Au début de chaque période, l'optimiseur considérera les retours des actifs du portefeuille sur une durée définie s'arrêtant au début de la période. Dans notre cas, cette période était généralement fixée à 6 mois : une optimisation effectuée le 1er juillet 2016, considère des données allant du 1er janvier 2016 au 31 juin 2016. Cette durée peut elle aussi être adaptée selon les besoins du portefeuille. En reprenant notre exemple, les actifs seront pondérés de façon identique du 1er juillet 2016 au 31 juillet 2016.



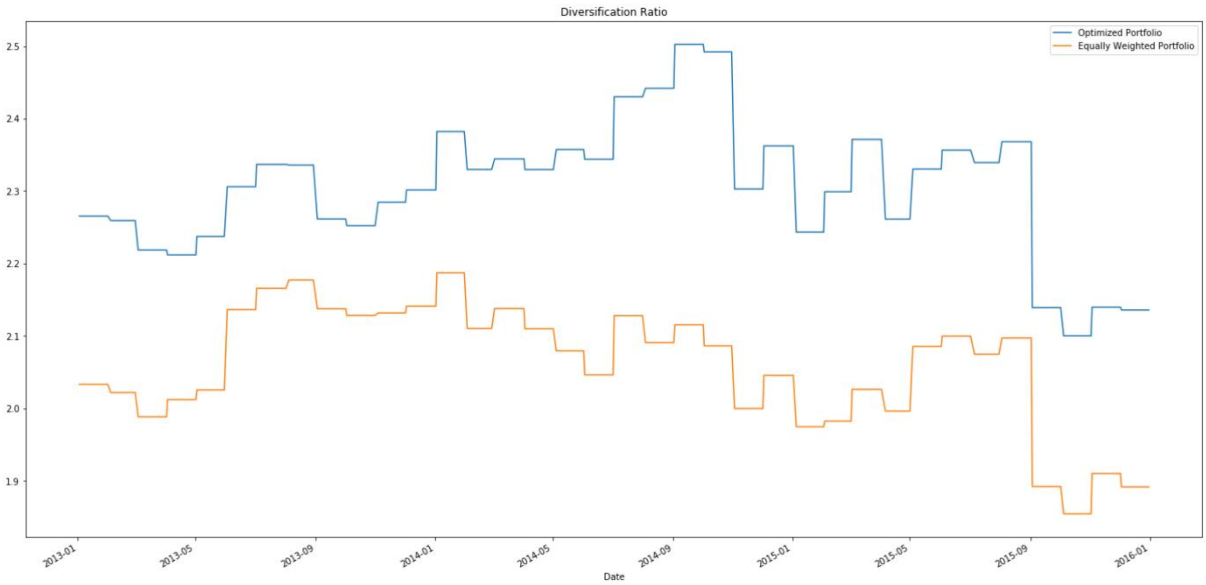
Évolution des poids en optimisant le ratio de diversification, pour 21 actifs du S&P500

V. Paramètres utilisés

1. Diversification ratio



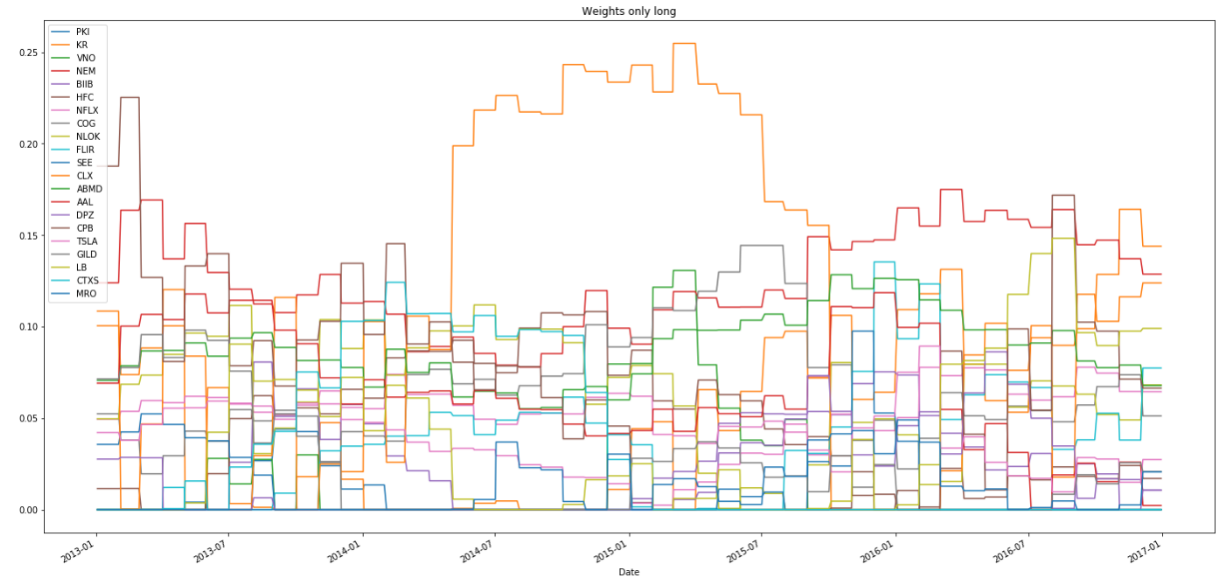
Évolution de la valeur du portefeuille (en %)



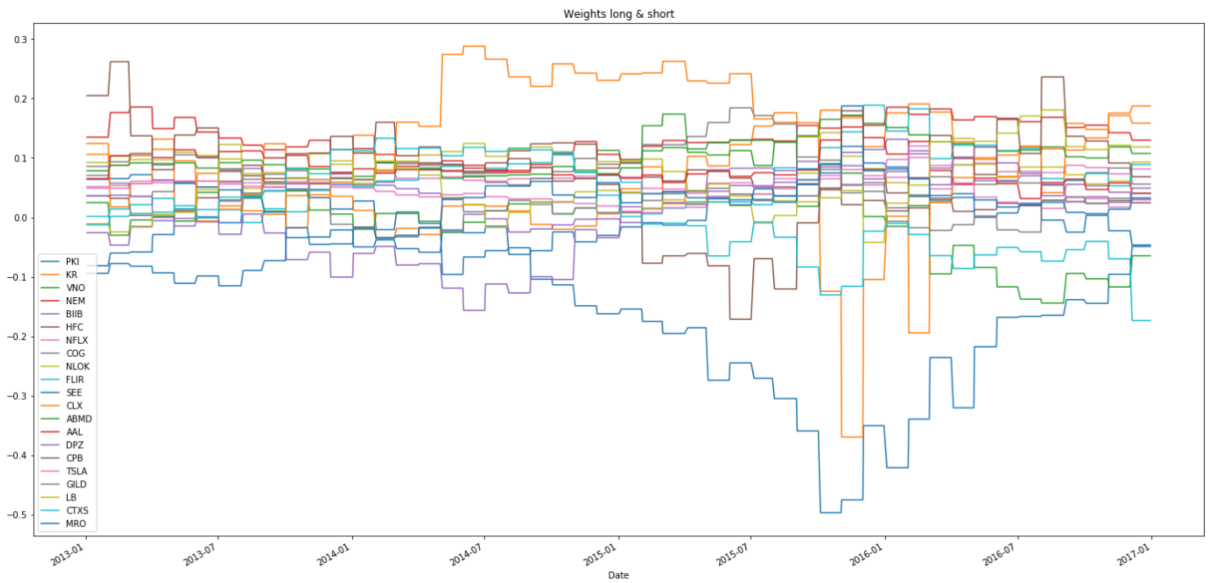
Ratio de diversification

Nous pouvons déjà observer la différence entre notre portefeuille optimisant le ratio de diversification et celui attribuant un poids égal à chaque actif. Les performances en termes de retour ne sont pas supérieures pour le portefeuille optimisé, ce qui n’est pas un souci, ici nous cherchons principalement à optimiser le risque lié au portefeuille. C’est ce qui est réalisé par l’optimiseur qui maximise le ratio de diversification, observable dans le second graphique, et donc plus élevé que pour le portefeuille à poids égaux. De cette manière, le risque du portefeuille est minimisé grâce à la diversification.

2. Long & short

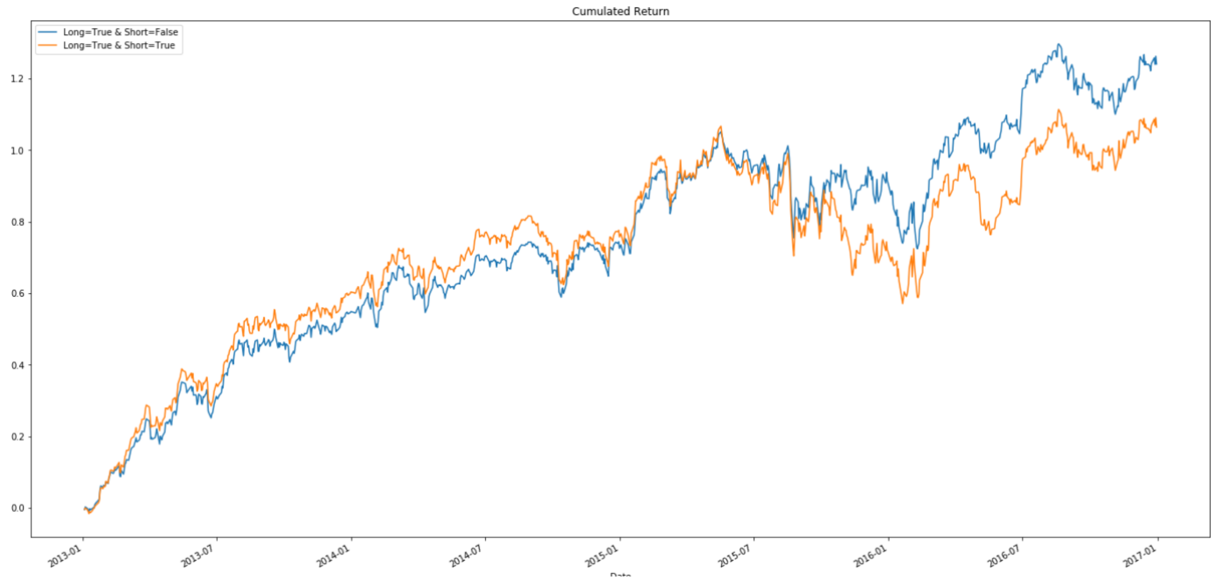


Poids dans une configuration uniquement long



Poids dans une configuration long & short

Dans la première configuration (long uniquement), l'optimiseur pourra uniquement acheter des actifs ou vendre ceux présent dans le portefeuille, ce qui résulte en des poids uniquement positifs. Dans la seconde configuration, il est cette fois possible de vendre des actifs à découvert (short), ce qui a pour conséquence de créer des poids négatifs pour certains actifs du portefeuille.



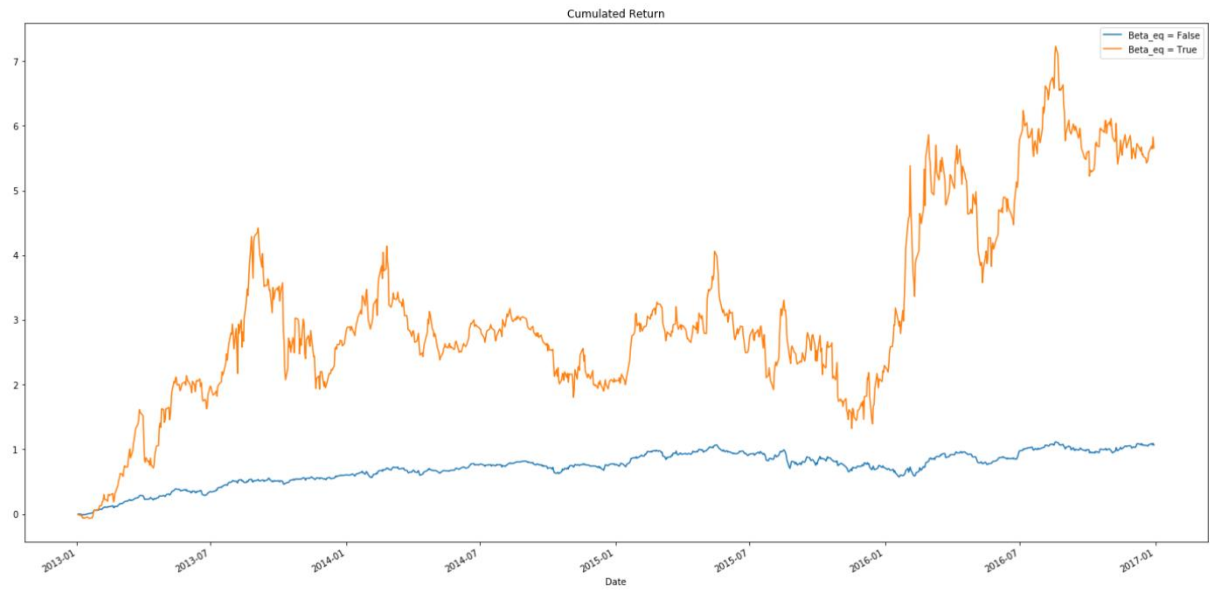
Évolution de la valeur du portefeuille (en %)

Les performances du portefeuille ne sont cependant pas fortement impactées par la modification de ce paramètre. Le marché étant globalement sur une période de croissance, cela peut justifier les 2 évolutions assez similaire.

3. Beta

Avec le paramètre β_{eq} activé, nous pouvons imposer à l'optimiseur la contrainte d'un β du portefeuille nul, et donc un portefeuille décorrélé du marché. Sur le graphique ci-dessous, on peut observer que le retour du portefeuille est plus important, mais sujet à des fluctuations beaucoup plus importantes.

En effet, la variance est plus importante. Par conséquent le ratio d'information (retour moyen divisé par la variance) est plus élevé pour β_{eq} = *False* (égal à 0.073) que pour β_{eq} = *True* (égal à 0.066).

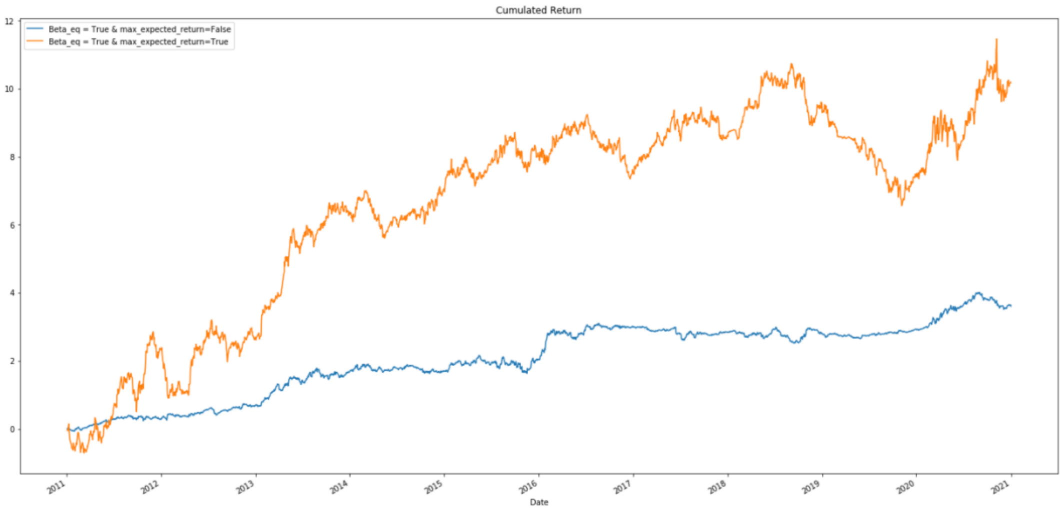


Évolution de la valeur du portefeuille (en %)

4. Return maximal

Jusqu’à maintenant notre portefeuille proposait une bonne gestion des risques, mais un rendement assez limité. En faisant le choix de privilégier le rendement quand celui est très prometteur. Sur la figure suivante, le rendement passe ainsi de +300% sur 10 ans à +1000%.

Même si la variance est plus élevée dans cette nouvelle configuration (0.058 contre 0.041), le ratio d’information est plus important (0.080 contre 0.066), ce qui justifie l’intérêt d’un trade-off pour améliorer les performances du portefeuille.



Évolution de la valeur du portefeuille (en %)

5. RSI1 & RSI2

On peut maintenant regarder l’influence de la prise en compte de l’indice RSI des deux façons présentées plus haut. On choisit par défaut des RSI à -10, ce qui signifie que l’on calcule l’indice à partir des données des 10 jours précédents.

Voici la comparaison entre la situation où le RSI n’est pas pris en compte en comparaison avec le RSI2 pris en compte, soit le RSI pris en compte dans la fonction objectif :

Sans prise compte du RSI

Prise en compte RSI2 (-10, 2)

On peut alors comparer l'effet de RSI1 et RSI2, soit RSI pris en compte comme contrainte ou comme objectif de minimisation :

Prise en compte du RSI1 (-10)

On voit alors que les deux indices RSI donnent des résultats d'optimisation très proches, si ce n'est similaires.

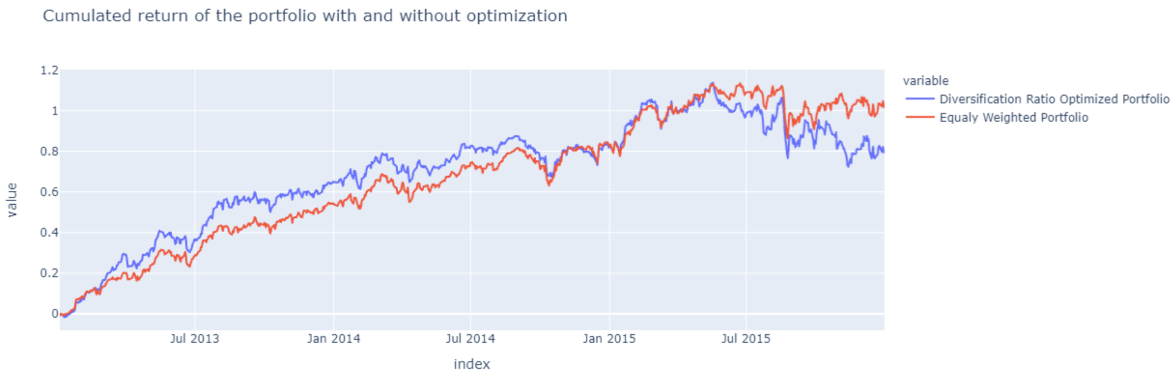
L'indice RSI permet alors, quelle que soit la façon dont on l'incorpore à notre optimisation, d'améliorer le *return* ainsi que le *diversification ratio*.

6. Turnover contraint et coûts de transition

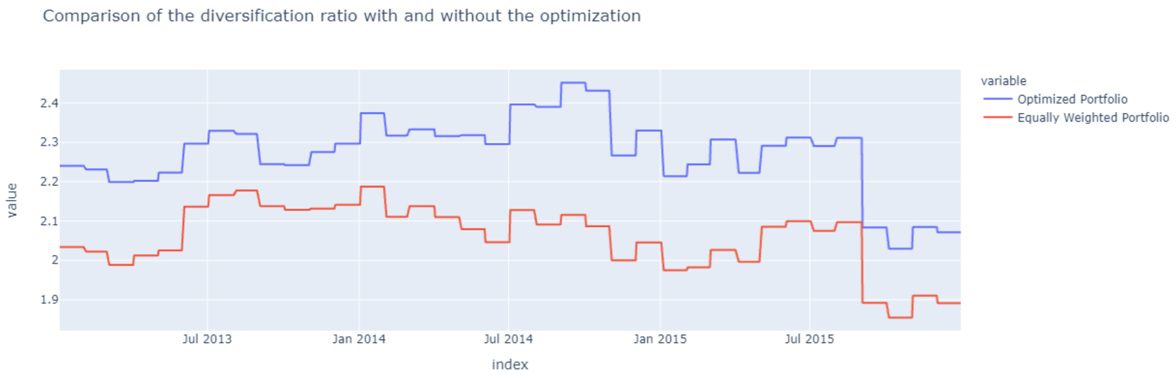
Nous choisissons pour nos transactions un pourcentage maximum de changement des allocations des actions : c'est le Maximum Turnover. La composition de notre portefeuille ne peut donc pas changer, pour chaque itération, de plus que la valeur fixée. Cette valeur se trouve entre 0 et 2 lorsque nous sommes dans la configuration *long/short*, et entre 0 et 1 lorsque l'on ne considère que du *long*.

On peut alors comparer différentes valeurs de Max_Turnover. Un premier chiffre qui nous paraît cohérent pour commencer est de 20%. On peut alors le faire varier jusqu'à 2 (en *long/short*). On peut également considérer plusieurs normes (q = 1 ou 2), et voir l'influence de chacune des deux méthodes.

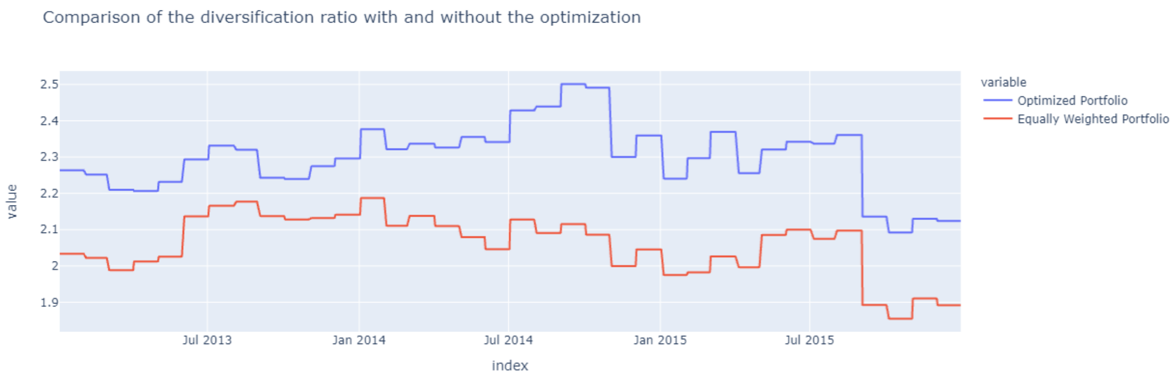
Cumulated return du portefeuille avec Max_Turnover = 0.2 en norme 2



Cumulated return du portefeuille avec Max Turnover = 0.5 en norme 2



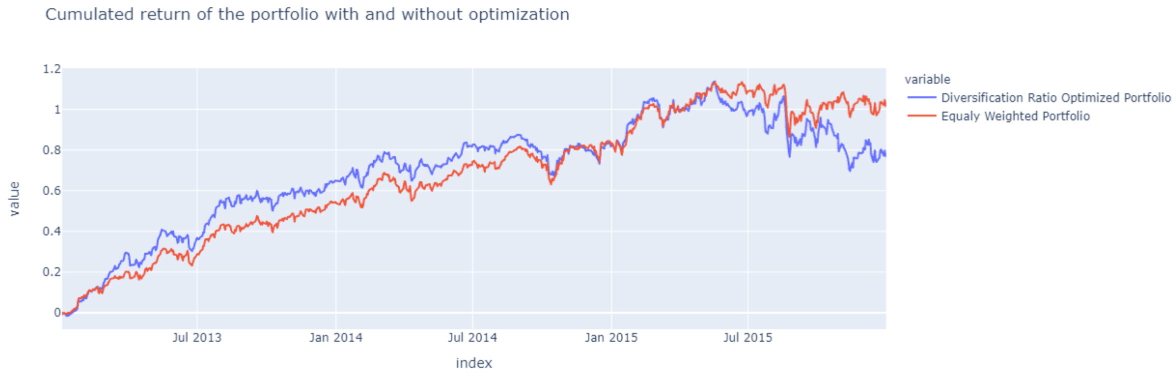
Diversification ratio avec Max_Turnover = 0.2 en norme 2



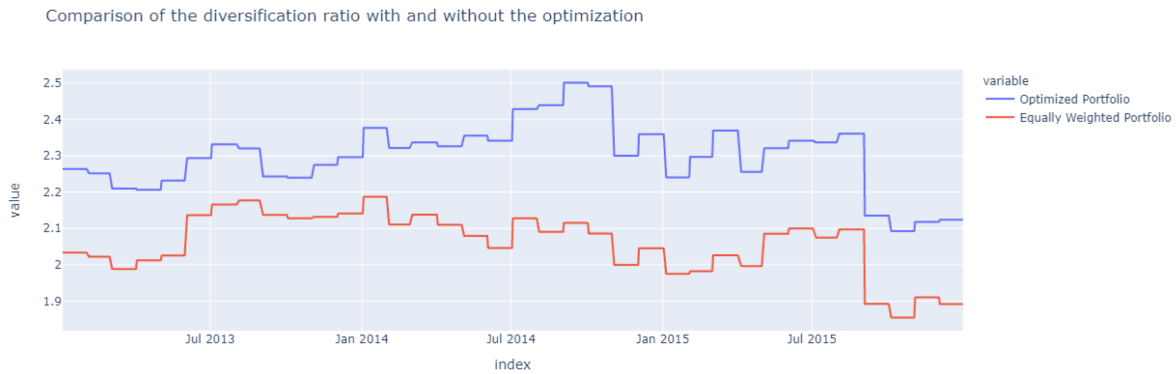
Diversification ratio avec Max_Turnover = 0.5 en norme 2

On voit alors que, lorsque l'on autorise une plus grande liberté de Turnover, on a naturellement un *cumulated return*, et un *diversification ratio* plus optimisé par rapport au portfolio avec des poids identiques sur toutes les actions.

Cependant, on constate qu'au-delà de Max_Turnover = 0.5, même si les allocations peuvent varier, les courbes de return sont quasiment similaires :

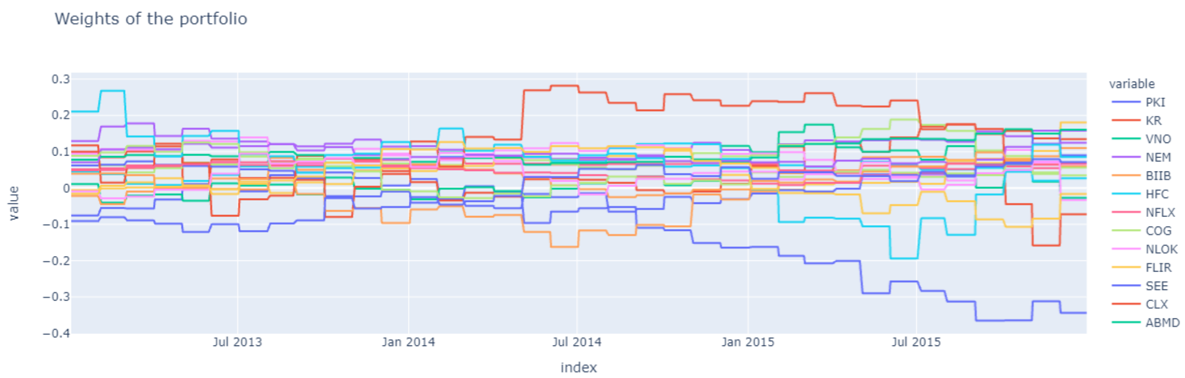


Cumulated return du portefeuille avec Max_Turnover = 1.3 en norme 2

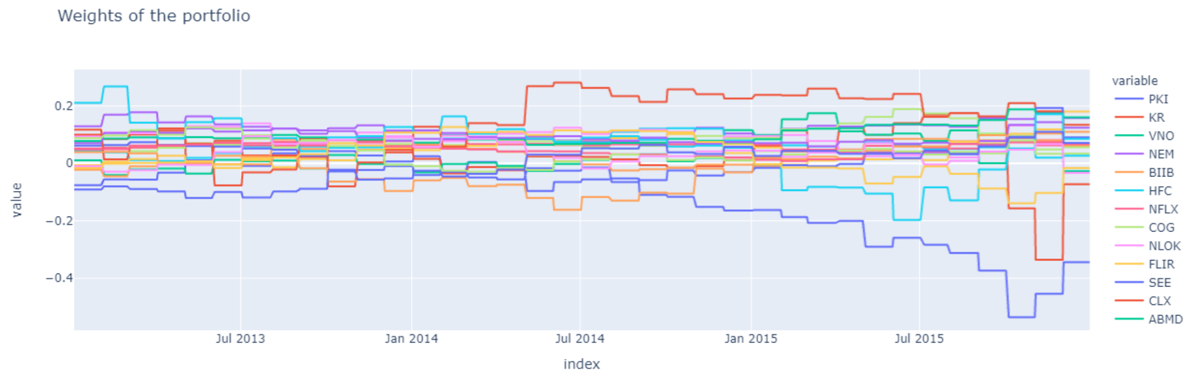


Diversification ratio avec Max_Turnover = 1.3 en norme 2

On peut alors conclure que <u>Max Turnover = 0.5 représente une valeur seuil</u>, après laquelle l'optimisation n'est pas meilleure au niveau du return, mais les allocations d'actions varient (très légèrement). Il nous semble alors plus ingénieux de ne pas dépasser cette valeur limite, car la dépasser autorise plus de transactions, ce qui n'est pas forcément optimal au niveau des *transaction costs*. **<u>On vérifie donc bien que le critère de Max_Turnover remplit sont objectif principal, à savoir : il nous permet d'obtenir des résultats aussi bons avec un nombre de transactions limitées.</u>**

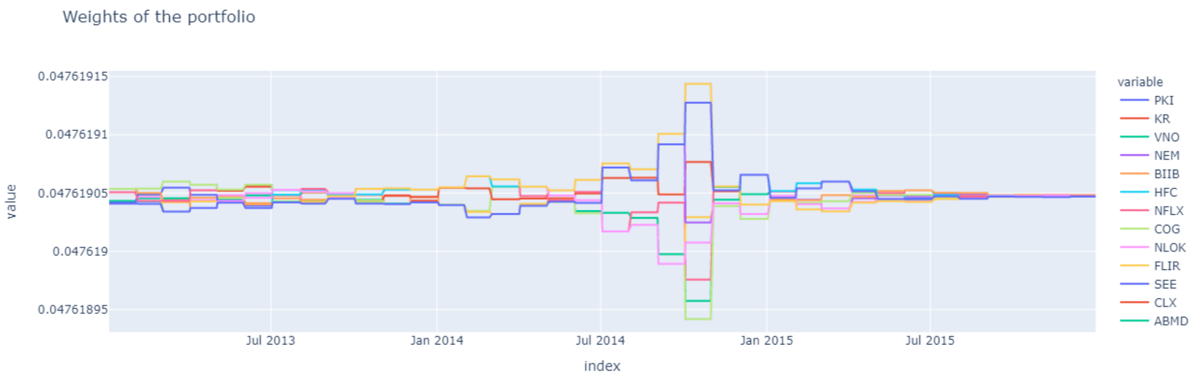


Weights of the portfolio for Max_Turnover = 0.5 en norme 2

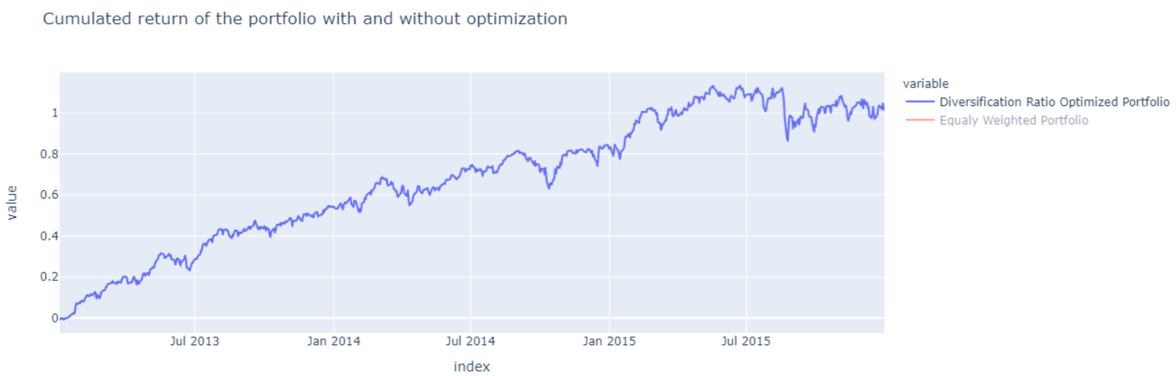


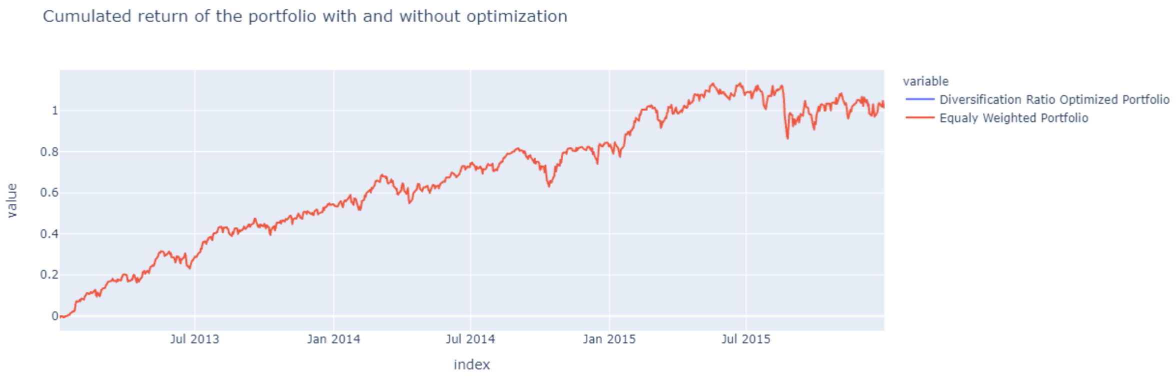
Weights of the portfolio for Max_Turnover = 1.3 en norme 2

On peut alors vérifier l'influence des *transaction costs* au-delà de cette valeur seuil. On s'attend à ce que l'augmentation du coût des transactions fasse réduire fortement les changements d'allocation. Avec des valeurs extrêmes, la répartition des poids se rapprocherait alors très fortement d'un portefeuille *equally weighted*, puisqu'on s'attend alors à une sorte de « valeur moyenne » pour chacun des poids. On lance la simulation pour Max_Turnover = 0.5, avec des Transaction Costs de 50 :



Weights of the portfolio for Max_Turnover = 0.5, Transaction Costs = 50 en norme 2





Cumulated return du portefeuille avec Max_Turnover = 0.5, Transaction Costs = 50 en norme 2

On retrouve en effet une simulation identique à celle du portefeuille *equally weighted*, et des poids tous proches d'une valeur moyenne qui est la même pour tous. Les désavantages de l'étude des Transaction Costs par rapport au Max_Turnover est qu'il est difficile de se rendre compte de l'influence des Transaction Costs lorsque l'on reste dans des valeurs réalistes (entre 0 et 1) : dans cet ordre de grandeur, les modifications ne sont pas significatives car l'importance de ce paramètre est faiblement pondéré dans la fonction objectif et donc dans le trade-off. Il est donc plus aisé d'optimiser nos résultats en passant par le Max_Turnover, même si, dans l'idée, ces deux paramètres traduisent la même contrainte (limiter le nombre de transactions d'actions).

7. Conclusion

Afin de conclure ce projet, nous pouvons maintenant regrouper nos différentes études des paramètres dans le but d'obtenir une simulation optimale sur tous les aspects. On peut donc retenir que notre optimisation est la plus efficace lorsque l'on combine à la fois la maximisation du diversification ratio et du *expected return*. On préfèrera également travailler en *long/short*, et utiliser la méthode du *rolling window* sur une période de 12 mois. La question se pose alors de la meilleure solution entre contraindre β_{eq} = 0 ou fixé Max_Turnover = 0.5.