

ingdevfi

Ingénieur en Développement dans la Finance

Kata Finance : Le MiniPricer

5 Votes

Introduction

Objectifs

Le premier objectif de ce Kata est identique à tous les Kata: S'améliorer dans la technique du TDD, par l'écriture de tests, l'écriture du code. Puis le refactoring des tests et du code à cause des changements de spécifications (complexification de l'algo, exception à la règle, besoin d'amélioration techniques induit,...).

Le deuxième objectif de ce kata est de vous donner une idée, certes sommaire mais idée tout de même, de comment peut fonctionner un pricer « On Demand » dans la finance de marché.

Enfin, pour certaines étapes on va voir les « limites » des tests unitaires, je vous laisse débattre de s'il faut trouver des moyens de les franchir ou pas. Et si oui de comment le faire.

Auditoire:

Pour réaliser ce Kata vous devez « maîtriser » un langage de programmation, ainsi qu'un framework de test. Il n'y a pas vraiment besoin de connaissance en finance, mais il est mieux d'y avoir une certaine sensibilité. Ce kata a aussi une relative complexité, il est bien d'en avoir fait des très simples avant de le commencer (ex: chercher Kata FizzBuzz sur le net).

Etapes:

Pour chaque étape, il y a un énoncé que j'ai essayé de faire le plus clair possible. Il y a aussi un paragraphe nommé: « A la fin de l'étape ». Il s'agit de quelques questions pouvant amener à une phase de refactoring ou non. Le but étant de challenger/réfléchir à ce que l'on a fait lors de l'étape. Il n'y a pas forcément de bonne réponse absolue à ces questions. Ne trichez pas et lisez le qu'une fois l'étape résolue

Etape 1: Extrapolation du prix simple

Enoncé:

Il faut créer un MiniPricer pouvant donner le prix dans le futur d'un instrument en fonction de:

- La date dans le futur du prix.
- Le prix actuel de l'instrument.

- La volatilité moyenne journalière en % de l'instrument.

Sachant que la volatilité moyenne journalière représente sa variation prix moyenne d'un jour à l'autre. En d'autres termes, si l'instrument vaut P à J , à $J+1$ il vaut $P \times (1+V/100)$ en moyenne. Où V est la volatilité journalière moyenne de l'instrument exprimée en %

Note: Pour cette étape on considère que la variation de prix de l'instrument est toujours égale à sa volatilité moyenne.

A la fin de l'étape:

Les marchés sont fermés le samedi, le dimanche, et les jours fériés, donc la variation de prix de l'instrument est nulle, ces jours là. Y avez vous pensé ?

Note: pour simplifier on peut ne considérer que les jours fériés à date fixe 01/01, 01/05, 08/05... et pas ceux comme le lundi de pentecôte, Lundi de Pâques... Toujours pour simplifier, on peut en considérer que les 3 premiers, donnée précédemment.

Etape 2: Le caractère aléatoire

Enoncé:

En fait, un instrument ne varie jamais exactement de sa volatilité moyenne tous les jours. Comme son nom le dit C'est une moyenne. De plus la valeur de la volatilité est une valeur absolue, donc le prix de l'instrument peut augmenter ou descendre de cette valeur.

Revoir la méthode de calcul du prix en considérant que chaque jour soit le prix augmente de la volatilité moyenne, soit le prix ne bouge pas, soit le prix baisse de la volatilité moyenne. Le choix entre les trois mouvements possibles devant être aléatoire.

A la fin de l'étape:

Comment avez vous fait pour écrire un test sachant que le résultat de la méthode à tester est plus ou moins aléatoire ?

Avez vous utilisé des tests sur les extremums (nb jours ouverts \times volatilité $\times -1 \geq$ Prix \geq nb jours ouverts \times volatilité $\times 1$) ?

Avez vous utilisé un faux (mock) générateur de nombre aléatoire en vérifiant le nombre de fois où il a été appelé par exemple ?

Etape 3: Monté-Carlo

Enoncé:

En fait faire un seul tirage aléatoire par jour n'est pas du tout satisfaisant d'un point de vue mathématique. Une meilleure stratégie est de créer un grand nombre de trajectoires de prix et d'en faire la moyenne. Une trajectoire de prix étant ce qu'on a fait à l'étape précédente. A savoir tous les jours on fait un tirage au sort de la variation du prix à appliquer, et ce pour chaque jour à considérer. C'est ce qu'on appelle utiliser la méthode de Monté-Carlo (grand nombre d'aléas en entrée d'un processus

connu et moyenne au final).

A la fin de l'étape:

Peut-on choisir le nombre de trajectoires ? Devrait-on d'après l'énoncé ?

Avez vous touché aux tests précédents ? Comment vérifier que l'algorithme a bien changé sans casser l'encapsulation ?

Avez vous mis en parallèle le calcul des trajectoires ? Si oui, doit/peut on le vérifier par un test ?

Etape 4: Le panier d'instruments

Énoncé:

Pour aller plus vite, on veut pricer tous les instruments d'un même panier en un coup. La méthode que l'on va appliquer est la suivante. On va pricer un instrument dit pivot selon la méthode précédente. Et grâce à la corrélation moyenne des autres instruments par rapport à cet instrument pivot, on calcul la variation de prix de ces instruments. La corrélation représente le lien entre la variation de prix de deux actifs. Elle est généralement exprimée de manière signée, mais en %. A savoir si la corrélation entre deux instruments vaut -1, cela veut dire que quand l'un monte de X%, l'autre baisse de X%

A la fin de l'étape:

Peut-on toujours pricer qu'un seul instrument ? Avez-vous gardé vos premiers tests ? Les avez vous refactorisés ?

Ce changement de spécification a-t-il engendré un changement de l'API de votre MiniPricer ? Est-ce nécessaire/bien ?

Etape 5: La précision sur demande

Avertissement:

Cette étape n'apporte pas grand chose en termes de TDD ou méthodologie de travail, il s'agit juste de changer l'algo de pricing pour plus de commodité de l'utilisateur. En gros c'est l'étape bonus. Ceux qui ne voudraient pas la faire, allez à la fin directement.

Énoncé:

On doit pouvoir choisir la précision du prix déterminé par le pricer. On doit pouvoir dire au pricer que l'on veut un prix précis à 0.01€ près.

Pour ceux qui sécheraient complètement sur l'algo à mettre en oeuvre voici une petite astuce: la précision du prix dépend du nombre de trajectoires calculées. Plus on a de trajectoires plus la précision est grande. Un moyen simple de le faire est de calculer un certains nombres de trajectoires, calculé la moyenne de la distance de chacune des trajectoires par rapport à la moyenne (écart type). De vérifier que l'écart type est en dessous de la précision demandée. Si ce n'est pas le cas

recommencer en ajoutant les trajectoires les nouvelles trajectoires aux anciennes et en recalculant le nouvel écart type et ainsi de suite. La différence de prix étant bornée entre les trajectoires, + ou - nb jour ouvrés x Volatilité aux extremums, plus le nombre de trajectoires augmente, plus la précision aussi.

Fin du kata

c'est le moment de faire le ménage !!!

Avez vous du code en commentaire ?

Avez vous des tests en commentaires ?

Avez vous des tests des inutiles ou qu'il ne passe pas ?

Avez vous des commentaires explicatifs ?

Combien avez vous fait de classe ?

Combien avez vous fait de méthode ?

Bravo !

📍 Nanterre, France

Partager :



+ Suivre

★ Aime

Soyez le premier à aimer cet article.

Sur le même thème

Algotrading chez les brokers -
Partie VI: L'implémentation
shortfall

L'implémentation shortfall Je n'ai
jamais vue de traduction du terme
implementation shortfall, je paye
une bière à celui qui trouve la
traduction officielle ;-)
Dans "Finance"



Algotrading chez les brokers -
Partie IV: Le VWAP
Dans "Finance"

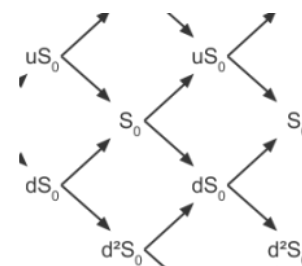
Entrez votre adresse mail

Inscrivez-moi

Construisez un site avec
WordPress.com

S'abonner à "ingdevfi"

Recevez les nouvelles
publications par mail.



Implementations de Cox Ross
Rubinstien
Dans "C#"

C#, Développement, Finance carlo, corrélation, finance, Kata, marché, monté, monté-carlo, montécarlo, pricer, TDD, trading, volatilité

← Brève: Un nouveau regard sur l'injection de dépendance

Implementations de Cox Ross Rubinstien →

Une réponse à “Kata Finance : Le MiniPricer”

ALAOUI Hasna 9 mars 2015 à 11 h 44 min

Bravo pour ton Blog Nicolas ! Hasna



★ Aimé par [1 personne](#)

Répondre