



DERIVATIVES

— FINANCIAL MARKETS —

PRÉSENTE :

PROJET PYTHON

La copie, l'envoi ou la reproduction de ce document est entièrement interdite.

Merci pour votre compréhension et excellente lecture !

DERIVATIVES

PROJET PYTHON

- 1) Créez une fonction permettant de calculer le prix d'un call ou d'un put du stock que vous avez choisi en utilisant le modèle de Black & Scholes. Votre fonction doit prendre les différents paramètres en entrée : S , K , r , q , σ ... et retourner le prix de l'option
- 2) Construisez une fonction permettant de calculer la volatilité implicite d'une option à partir de ses caractéristiques. Il existe plusieurs méthodes lorsque l'on cherche à déterminer la volatilité implicite d'une option : on distingue notamment la méthode Newton, la bisection, la méthode de la sécante, OLS (régression linéaire) ou même des simulations de MonteCarlo. La plus simple est la méthode de Newton, que vous pouvez implémenter vous-même très rapidement.

En Bonus, vous pouvez également essayer de trouver la racine de la fonction via d'autres méthodes, et comparer les différences : le module `scipy` permet notamment d'en utiliser plusieurs facilement, sans avoir à développer vous-même chaque méthode de résolution.

- 3) Maintenant que nous avons de quoi analyser et calculer les valeurs théoriques des options, nous devons trouver des données. Yahoo finance permet également de récupérer des données d'options. Attention, récupérer ces données est un peu plus « complexe » que les simples cours historiques. Regardez la documentation et notamment la méthode « Option-chain » d'un objet `yf Ticker`.

Cette question peut sembler facile mais est assez importante : elle ressemble en fait à une part quotidienne de votre futur travail si vous êtes amenés à programmer dans vos stages /summer. Il vous faudra impérativement bien lire et comprendre la documentation, afin d'utiliser correctement/les bonnes fonctions.

- 4) Visualisez le contenu des dataframe que vous obtenez via les méthodes `.get_call` et `.get_put` : on observe une colonne dédiée à la volatilité implicite. Utilisons nos fonctions créées précédemment pour comparer nos résultats avec ces données. Nous allons d'abord pour ce faire n'utiliser que les options relativement liquides et restreindre notre analyse. On ne peut faire de déduction de données non nettoyées ou impropres à être analysées, c'est exactement le but de cette questions.

Il vous faut sélectionner les options relativement liquides : quels critères pouvez-vous utiliser ? Visualisez le dataframe et ses colonnes, et formulez des hypothèses. Appliquez ces critères à votre dataframe et assignez le résultat à une nouvelle variable.

Tips 1 : la colonne 'volume' devrait vous être utile. Les options très loin de la monnaie (OTM comme ITM) sont très peu liquides et sans doute à exclure

- 5) Un exercice intéressant est désormais de visualiser les volatilités implicites contenues dans votre dataframe. Sélectionnez une maturité, puis afficher la volatilité implicite en fonction des différents strike. Quelle forme remarquez-vous et comment l'expliquer ?
- Vous pouvez répéter l'opération avec plusieurs maturité : il s'agira cette fois de se rendre compte de la term structure de la volatilité. Vous pouvez aussi visualiser la vol selon les maturités, en choisissant arbitrairement un strike. Pour visualiser cette term structure.
- 6) Maintenant que nous avons des données « propres », nous pouvons procéder à une analyse et calculer nous-mêmes les volatilités implicites des options.
- Une solution facile consiste à ajouter une nouvelle colonne au dataframe, puis de remplir pour chaque ligne (pour chaque option) par la volatilité implicite calculée via nos fonctions précédentes. **Attention : l'intérêt de cette question est d'utiliser les fonctionnalités de la librairie pandas. Il n'est pas question de faire une boucle for sur l'ensemble des lignes et de remplir notre colonne cellule par cellule.**
- Tips : regardez dataframe.apply() est un bon point de départ.
- Pensez à bien utiliser le prix de l'option figurant dans le dataframe et ne pas le calculer vous-même. De nombreuses autres informations (notamment la date d'expiration) figurent également dans ce dataframe. Il vous faut également le prix Spot de l'action, les fonctions de l'exercice 1 vont peut-être vous être utiles.
- 7) Vous devez normalement désormais avoir un dataframe contenant le prix de l'option, sa volatilité implicite « réelle » et sa volatilité implicite que vous avez calculé vous-même. Créez une nouvelle colonne, par exemple « comparaison » et réaliser la comparaison entre les 2 volatilités implicites. Pourquoi selon-vous constate on des écarts ?
- 8) Utilisez votre fonction de calcul du prix d'une option et calculer le prix « théorique » de chaque ligne de votre dataframe d'options. Pensez à utiliser à nouveau dataframe.apply() et à éventuellement adapter votre fonction de calcul de BS . Si vous n'y arrivez pas, vous pouvez toujours faire cellule par cellule via une boucle for. Comparez votre prix théorique et le prix indiqué par Yahoo finance.