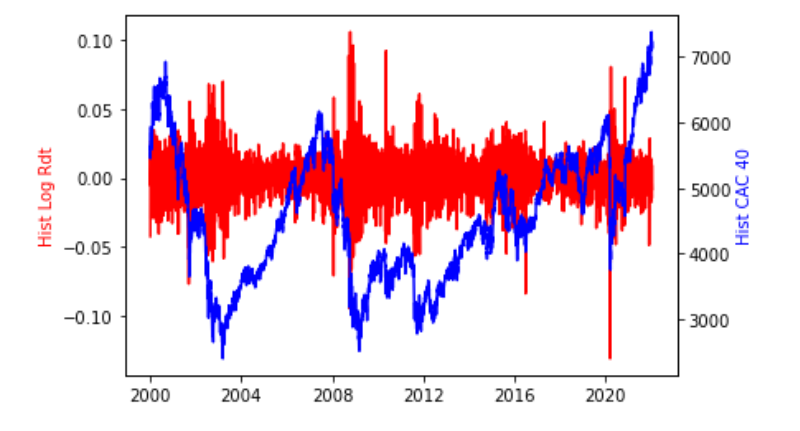
* 1. Import des données (Date, Close)
  2. Analyse préalable et nettoyage si nécessaire (valeurs NaN)
  3. Calcul du log-rendement
  4. Graphique : y1 – cours & y2 – log-rendement vS x – Date



* 1. Découpage Apprentissage / Test :

Apprentissage : **15 octobre 2008 - 26 juillet 2022**

Test : **27 juillet 2022 - 11 juin 2024**

* 1. Statistiques descriptives sur les 2 jeux de données

1. **VaR non paramétrique**
   1. Ecrire une fonction calculant la VaR historique d’un ensemble de log-rendements  
      *🡺 VaR\_Hist(x, alpha)*
   2. Calculer la VaR historique sur base d’apprentissage pour alpha = 99%.
   3. Ecrire une fonction calculant la VaR historique bootstrap d’un ensemble de log-rendements et donnant un IC de niveau alpha\_IC de cette VaR  
      *🡺 VaR\_Hist\_Bootstrap(x, alpha, B, alpha\_IC)*
   4. Calculer la VaR historique bootstrap et l’IC associé à 90% sur base d’apprentissage pour alpha = 99%.
   5. Calculer le nombre d’exceptions sur base de test associées à la VaR historique calculée en 1.b.
   6. Comparer **statistiquement** ce % d’exceptions avec le niveau de risque attendu. (*Aide : on pourra par exemple passer par un IC de binomiale)*
2. **VaR gaussienne**
   1. Ecrire une fonction calculant la VaR gaussienne d’un ensemble de log-rendements  
      *🡺 VaR\_Gauss(x, alpha)*
   2. Calculer la VaR gaussienne sur base d’apprentissage pour alpha = 99%.
   3. Faire une validation ex-ante (analyses graphiques, QQ-plot, etc.)
   4. Calculer la VaR gaussienne à 10j par la méthode du scaling
   5. Calculer la VaR gaussienne à 10j par méthode de diffusion d’un actif.  
      On pourra considérer :
      1. comme la valeur du cours de clôture à la dernière date de l’échantillon d’apprentissage
      2. et les paramètres estimés en 2.b
   6. Ecrire une fonction calculant la VaR gaussienne à 1j d’un ensemble de log-rendements en surpondérant les observations les plus récentes selon la méthode EWMA (Exponential Weighting Moving Average)  
      *🡺 VaR\_Gauss\_EWMA(x, alpha, lambda)*
   7. Pour lamba = 0.9, 0.95 et 0.99 :
      1. Calculer les paramètres et
      2. Calculer les VaR gaussienne EWMA sur base d’apprentissage pour alpha = 99%.
      3. Calculer le nombre d’exceptions sur base de test.
3. **VaR skew-Student**
   1. Ecrire une fonction permettant d’estimer les paramètres d’une loi de Skew Student par maximum de vraisemblance.

On procédera par étapes : fonction de densité, fonction de log-vraisemblance, fonction d’optimisation.

On pourra utiliser le package *scipy.optimize* et la fonction *minimize*.

* 1. Estimer les paramètres de loi Skew Student sur base d’apprentissage.
  2. Faire une validation ex-ante par QQ-plot.
  3. Comparer la qualité de fit entre loi gaussienne et loi de skew Student par analyse graphique.
  4. Calculer la VaR Skew Student sur base d’apprentissage pour alpha = 99%.

1. **Expected Shortfall**
   1. Calculer l’ES empirique associé à la mesure faite en 1.b sur la base des données observées
   2. Calculer
      1. L’ES empirique associé à la mesure faite en 2.b sur la base des données observées
      2. L’ES théorique associé à la mesure faite en 2.b sur la base de la loi calibrée
   3. Calculer
      1. L’ES empirique associé à la mesure faite en 3.e sur la base des données observées
      2. L’ES théorique associé à la mesure faite en 3.e sur la base de la loi calibrée