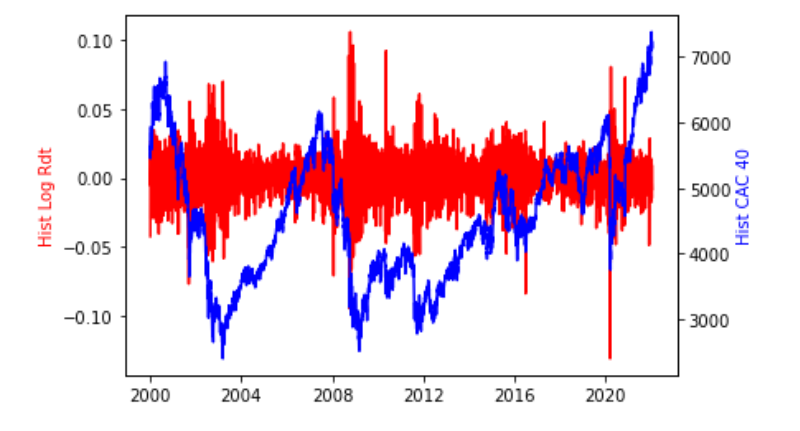
* 1. Import des données (Date, Close)
  2. Analyse préalable et nettoyage si nécessaire (valeurs NaN)
  3. Calcul du log-rendement
  4. Graphique : y1 – cours & y2 – log-rendement vS x – Date



* 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
2. **Protocole de backtesting**
   1. Proposer un protocole de backtesting selon les modalités suivantes :
      1. Le modèle initial est calibré sur la période d’apprentissage
      2. Le modèle est backtesté ex-post chaque jour suivant la période de calibrage
      3. Mise en place d’une mesure indiquant la nécessité ou non de recalibrer le modèle
   2. Mettre en place ce protocole sur les données de test

**Attention : la TVE s’intéresse aux queues de distribution droites (maximum, excès par valeurs supérieures). Penser à convertir les données avant de débuter.**

**On pourra utiliser pour les VaR TVE les lois *genextreme*, *gumbel\_r* et *genpareto* du package *scipy.stats* et les méthodes *fit*, *pdf*, *ppf*.**

1. **VaR TVE : Approche Maxima par bloc**
   1. Déterminer une taille de bloc *s* et construire un échantillon de maxima sur la base d’apprentissage.
   2. Tracer le Gumbel plot pour juger de l’hypothèse (i.e. GEV vs EV).
   3. Estimer les paramètres de loi GEV (ou EV si pertinent).
   4. Faire une validation ex-ante (analyses graphiques, QQ-plot, etc.).
   5. Calculer la VaR TVE par MB sur base d’apprentissage pour alpha = 99%.
2. **VaR TVE : Approche Peak over threshold**
3. Ecrire une fonction permettant d’obtenir le *mean excess* plot.
4. Déterminer *u* par analyse graphique.

*Attention : ne pas considérer ni les valeurs négatives de u, ni les valeurs trop extrêmes, au risque d’écraser l’échelle visuelle du graphique.*

1. Estimer les paramètres de loi GPD.
2. Faire une validation ex-ante (analyse graphiques, QQ-plot, etc.).
3. Calculer la VaR TVE par PoT sur base d’apprentissage pour alpha = 99%.
4. Proposer un protocole permettant de calibrer *u* de manière automatique, et le mettre en œuvre.