

데이터 생성 (시뮬레이션)

1. market_volatility_proxy

시간이 지남에 따라 증가하는 값을 가집니다. 이는 마치 시장 불안정성이 점점 커지는 상황을 가정한 것입니다.

2. returns

수익률 데이터는 `np.random.normal` 함수를 사용하여 생성합니다.

이때 `scale` 인자로 `market_volatility_proxy`를 전달하여, `market_volatility_proxy`가 커질수록 `returns`의 표준편차(변동성)도 커지도록 합니다. 이 부분이 이분산성(heteroscedasticity)을 시뮬레이션 합니다.

3. lagged_returns

VaR 모델의 독립변수로 사용될 과거 수익률입니다.

분위수 회귀 모델 구축

`statsmodels.formula.api.quantreg` 함수를 사용하여 분위수 회귀 모델을 정의합니다. `formula` 인자로 종속변수 ~ 독립변수1 + 독립변수2 형식으로 모델을 지정합니다.

VaR는 수익률 분포의 특정 하위 분위수에 해당합니다. 예를 들어, 99% VaR는 수익률 분포의 하위 1% 지점입니다. 따라서 `model.fit(q=0.01)`을 호출하여 수익률의 0.01 분위수(즉, 1분위수)에 해당하는 회귀 계수를 추정합니다.

이때 분위수 회귀는 손실(음의 수익률)이 과대 예측될 때 더 큰 페널티를 부과하여, 모델이 해당 0.01 분위수 아래로 실제 수익률이 내려가는 경우를 최소화하도록 학습합니다. 즉, 수익률의 꼬리 부분을 정확하게 모델링하는 데 집중합니다.

VaR 값 예측

`result_qr_var.predict(data)`를 호출하여 각 시점의 설명 변수(과거 수익률, 시장 변동성 프록시)를 기반으로 수익률의 0.01 분위수 값을 예측합니다.

VaR는 통상적으로 양수로 손실액을 표현하므로, 예측된 수익률 분위수 값에 `-`를 취하여 `predicted_var`를 계산합니다. (예: 0.01 분위수가 -0.02라면, VaR는 0.02 즉 2% 손실)

시각화

실제 일일 수익률과 추정된 VaR 라인을 함께 그립니다.

VaR 라인은 수익률의 하단 경계선 역할을 합니다. 이분산성 때문에 시간이 지날수록 실제 수익률의 변동폭이 커지고, 이에 따라 VaR 라인도 점점 더 아래쪽으로 내려가면서 변동 폭을 크게 가져가는 것을 볼 수 있습니다.

VaR 위반 지점(실제 수익률이 VaR 라인 아래로 내려간 경우)을 보라색 점으로 표시하여 시각적으로 모델의 성능을 평가할 수 있습니다.

간단한 백테스팅

num_exceedances를 통해 실제 수익률이 추정된 VaR를 초과(손실 발생)한 횟수를 계산합니다.

exceedance_rate는 전체 기간 동안 VaR 위반이 발생한 비율을 나타냅니다.

이론적으로, 99% VaR는 1%의 위반율을 가져야 합니다. 즉, 1000일 중 약 10일 정도 VaR 위반이 발생해야 합니다. 이 예시에서는 시뮬레이션 데이터이므로, 실제 위반율이 기대 위반율에 가깝게 나오면 모델이 데이터를 잘 포착했다고 볼 수 있습니다.