평균 변화율: 기하 평균, 로그 변화율

데이터 분석, 특히 금융이나 경제 분야에서는 값의 절대적인 크기보다 시간에 따른 **변화율(Rate of Change)**이 더 중요한 경우가 많습니다. 여러 기간에 걸친 평균적인 변화율을 계산할 때, 일반적인 산술 평균은 왜곡된 결과를 낳을 수 있습니다. 이때 사용되는 개념이 기하 평균과 로그 변화율입니다.

기하 평균 (Geometric Mean)

- n개의 양수 값들을 모두 곱한 후, n제곱근을 취한 값입니다.
- 주로 **비율(ratio)이나 성장률(growth rate)의 평균**을 계산하는 데 사용됩니다.
- 각 기간의 성장률이 곱셈으로 연결되기 때문에, 덧셈을 기반으로 하는 산술 평균보다 시간에 따른 평균 변화율을 더 정확하게 나타냅니다.
- **공식**: $G = (x_1 * x_2 * ... * x_n)^(1/n)$
- **예시**: 2년 동안 주식 수익률이 각각 +50%, -50%일 때, 산술 평균은 (50 - 50) / 2 = 0%이지만, 실제로는 100만원 -> 150만원 -> 75만원이 되어 원금 손실이 발생합니다. 기하 평균은 이 상황을 정확히 반영합니다.

로그 변화율 (Logarithmic Rate of Return)

- 특정 기간 동안의 변화율을 로그를 이용하여 표현한 것입니다. $ln(현재 가격 / 과거 가격)$으로 계산됩니다.
- **특징**:
    1. **시간 가산성(Time Additivity)**: 여러 기간의 로그 변화율을 단순히 더하면 전체 기간의 총 로그 변화율이 됩니다. (e.g., 1월의 로그수익률 + 2월의 로그수익률 = 1~2월의 총 로그수익률). 이 특징 덕분에 다루기가 매우 편리합니다.
    2. **대칭성**: +10% 상승과 -10% 하락의 절대값이 다르게 계산되는 일반 수익률과 달리, 로그 변화율은 그 크기가 거의 대칭적입니다.
    3. **정규분포 근사**: 주가와 같은 금융 데이터의 수익률은 로그를 취했을 때 정규분포에 더 가까워지는 경향이 있습니다. 이는 통계적 모델링에 유리하게 작용합니다.

적용 가능한 상황

- **기하 평균**:
    - 여러 해에 걸친 연평균 성장률(CAGR, Compound Annual Growth Rate)을 계산할 때.
    - 투자 포트폴리오의 기간별 수익률에 대한 평균 수익률을 계산할 때.
    - 서로 다른 스케일을 가진 여러 지표들의 평균적인 수준을 비교할 때 (e.g., 데이터베이스 벤치마크 성능 평가).
- **로그 변화율**:
    - 금융 시계열 데이터(주가, 환율 등)의 수익률을 계산하고 분석할 때. (가장 표준적인 방법)
    - 시계열 모델(ARIMA 등)의 입력 변수로 사용하여 시계열을 안정적인(stationary) 형태로 변환할 때.
    - 금융 파생상품 가격 결정 모델(e.g., 블랙-숄즈 모델)에서 기초자산의 변동성을 추정할 때.

예제 데이터 생성

```
import pandas as pd
import numpy as np
from scipy.stats.mstats import gmean

# 주가 데이터 예시
data = {'price': [100, 110, 105, 120, 90]}
df = pd.DataFrame(data)
```

기하 평균 (Geometric Mean)

- **문제**: 4개 기간 동안의 성장률에 대한 평균 성장률을 계산합니다.
- **주의사항**
  - 기하 평균은 데이터에 0 또는 음수가 있으면 계산할 수 없습니다.
  - 성장률을 계산할 때는 (현재값 / 이전값) 형태로 계산된 비율 값들의 기하 평균을 구한 후, 1을 빼서 최종 평균 성장률을 얻습니다.

```
# 각 기간의 성장률이 아닌, '비율'을 계산 (1 + 성장률)
df['growth_ratio'] = df['price'] / df['price'].shift(1)
print("--- Growth Ratios ---")
print(df)
'''
--- Growth Ratios ---
   price  growth_ratio
0    100           NaN
1    110      1.100000
2    105      0.954545
3    120      1.142857
4     90      0.750000
'''

# 결측치(첫 행)를 제외하고 기하 평균 계산
valid_ratios = df['growth_ratio'].dropna()

# 1-1. Scipy 사용
geo_mean_ratio_scipy = gmean(valid_ratios)
avg_growth_rate_scipy = geo_mean_ratio_scipy - 1
print(f"\nAverage Growth Rate (Scipy): {avg_growth_rate_scipy:.4%}") # -2.5996%

# 1-2. 수동 계산
# (1.1 * 0.9545 * 1.1428 * 0.75)^(1/4) - 1
geo_mean_ratio_manual = np.prod(valid_ratios)**(1/len(valid_ratios))
avg_growth_rate_manual = geo_mean_ratio_manual - 1
print(f"Average Growth Rate (Manual): {avg_growth_rate_manual:.4%}") # -2.5996%

# 최종 값 확인: 100 * (1 - 0.026) * (1 - 0.026) * (1 - 0.026) * (1 - 0.026) ~= 90
final_value_check = df['price'].iloc[0] * (1 +
avg_growth_rate_manual)**len(valid_ratios)
print(f"Final value check: {final_value_check:.2f} (실제 최종값:
{df['price'].iloc[-1]}) ")
# Final value check: 90.00 (실제 최종값: 90)
```

- **결과 해석**
  - 4개 기간 동안의 평균 성장률은 약 `-2.60%`임을 알 수 있습니다.
  - 산술 평균을 사용하면 `(10% - 4.55% + 14.28% - 25%) / 4 = -3.82%`로 다른 결과가 나옵니다.
  - 기하 평균으로 계산된 성장률을 초기값에 복리로 적용하면 최종값(90)과 일치하여, 평균적인 변화를 더 정확하게 나타냄을 확인할 수 있습니다.

로그 변화율 (Log Return)

- **용도**: 시계열 데이터의 연속적인 변화율을 계산합니다.
- **주의사항**
  - 로그 변화율은 일반적인 산술 변화율(`(현재-과거)/과거`)과 값이 약간 다릅니다.
  - 변화율이 작을 때는 두 값이 거의 비슷하지만, 변화율이 클수록 차이가 커집니다.

```
# ln(P_t / P_{t-1}) = ln(P_t) - ln(P_{t-1})
df['log_return'] = np.log(df['price']) - np.log(df['price'].shift(1))
# 또는 df['log_return'] = np.log(df['price'] / df['price'].shift(1))

# 2-2. 일반 산술 변화율 계산 (비교용)
df['arithmetic_return'] = df['price'].pct_change()

print("\n--- Log Return vs Arithmetic Return ---")
print(df[['price', 'arithmetic_return', 'log_return']])
'''
--- Log Return vs Arithmetic Return ---
   price  arithmetic_return  log_return
0    100                NaN         NaN
1    110           0.100000    0.095310
2    105          -0.045455   -0.046520
3    120           0.142857    0.133531
4     90          -0.250000   -0.287682
'''

# 2-3. 시간 가산성 확인
total_log_return = df['log_return'].sum()
total_period_return_from_log = np.exp(total_log_return) - 1
actual_total_return = (df['price'].iloc[-1] / df['price'].iloc[0]) - 1

print(f"\nTotal Log Return: {total_log_return:.4f}") # -0.1054
print(f"Total return calculated from sum of log returns:
{total_period_return_from_log:.2%}") # -10.00%
print(f"Actual total return over the period: {actual_total_return:.2%}") # -10.00%
```

- **결과 해석**
  - 로그 변화율(`log_return`)과 산술 변화율(`arithmetic_return`)은 비슷한 값을 갖지만 완전히 같지는 않습니다.
  - 가장 중요한 특징은 **시간 가산성**입니다. 각 기간의 로그 변화율을 모두 더한 값(`total_log_return`)에 지수 함수(`np.exp`)를 취하면 전체 기간의 총 변화율(`-10.00%`)을 정확하

게 복원할 수 있습니다. 이는 산술 변화율에는 없는 편리한 속성입니다.

장단점

개념	장점	단점
기하평균	시간에 따른 평균 변화율(성장률, 수익률)을 정확하게 계산함. 이상치에 덜 민감함.	데이터에 0 또는 음수가 있으면 사용할 수 없음. 산술 평균보다 계산이 복잡하고 직관적이지 않을 수 있음.
로그변화율	**시간 가산성**: 여러 기간의 수익률을 쉽게 합산하고 분석할 수 있음. 분포가 정규분포에 가까워져 통계적 모델링에 용이함.	수익률 자체의 의미가 산술 수익률보다 직관적이지 않음. 여러 자산으로 구성된 포트폴리오의 전체 수익률을 계산할 때는 개별 자산의 로그 수익률을 단순히 합산할 수 없음.