

# DIVERSIFIED PORTFORLIO

## CONTENTS

- 1) 데이터 수집
- 2) 수익률 및 통계 계산
- 3) 샤프 비율 계산 함수
- 4) 다양한 가중치 조합 계산
- 5) 최적 포트폴리오 도출
- 6) 시각화 ① (리스크-수익률-샤프)
- 7) 시각화 ② (가중치 VS 샤프 비율)

# DATA

선택한 3개의 회사는 국내 대표 식품회사

식품 산업에 대한 개인적인 관심과 함께

안정적인 소비재 산업군이라는 점에서 포트폴리오 분석에 적합하다고 판단

- 풀무원 : 건강식·친환경 이미지로 특화, 지속 가능한 소비 트렌드에 부합
- Cj제일제당 : 국내 식품 업계 선도 기업, 다양한 가공식품 브랜드 보유
- 오뚜기 : 가정간편식 분야에서 강세, 보수적인 재무 운영으로 안정성 확보

# DATA

```
In [8]: # =====
# 1. 데이터 수집
# =====
tickers = ['097950.KQ', '007310.KS', '017810.KQ']
names = ['CJ제일제당', '오뚜기', '풀무원']
start = '2023-11-01'
end = '2024-05-26'

all_data = []

for ticker, name in zip(tickers, names):
    df = yf.download(ticker, start=start, end=end)[['Close']] |
    df['Company'] = name
    df.reset_index(inplace=True)
    all_data.append(df)

merged = pd.concat(all_data, ignore_index=True)

pivot_df = merged.pivot_table(index='Date', columns='Company', values='Close')
pivot_df.dropna(inplace=True)

pivot_df.head()
```

분석 기업의 티커와 기업이름을 리스트로 정의  
기간은 6개월로 설정  
파이낸스에서 종가만 다운로드  
각 데이터를 합병하여 데이터프레임을 만듦

Out [8] :

Ticker	007310.KS	017810.KQ	097950.KQ
Company	오뚜기	풀무원	CJ제일제당
Date			
2023-11-01	372115.78125	9907.037109	278695.00000
2023-11-02	365905.90625	9966.066406	276766.28125
2023-11-03	370682.75000	9975.905273	279177.15625
2023-11-06	366861.28125	10025.095703	288820.56250
2023-11-07	361606.75000	10103.800781	283998.87500

## 수익률 및 통계 계산

```
In [9]: # =====  
# 2. 수익률 및 통계 계산  
# =====  
returns = pivot_df.pct_change().dropna()  
mean_returns = returns.mean()  
std_returns = returns.std()  
cov_matrix = returns.cov()  
corr_matrix = returns.corr()
```

pct\_change()를 통해 일별 수익률을 계산하고 returns에 저장  
일별수익률에 대한 평균, 분산, 표준편차, 상관관계를 불러옴

## 수익률 및 통계 계산

```
In [13]: mean_returns
```

```
Out[13]: Ticker      Company
007310.KS  오뚜기      0.001233
017810.KQ  풀무원      0.003664
097950.KQ  CJ제일제당    0.001894
dtype: float64
```

```
In [11]: std_returns
```

```
Out[11]: Ticker      Company
007310.KS  오뚜기      0.013213
017810.KQ  풀무원      0.022379
097950.KQ  CJ제일제당    0.016963
dtype: float64
```

```
In [12]: cov_matrix
```

```
Out[12]:
```

	Ticker	007310.KS	017810.KQ	097950.KQ
	Company	오뚜기	풀무원	CJ제일제당
	Ticker	Company		
	007310.KS	오뚜기	0.000175	0.000085
	017810.KQ	풀무원	0.000085	0.000501
	097950.KQ	CJ제일제당	0.000070	0.000138

```
In [14]: corr_matrix
```

```
Out[14]:
```

	Ticker	007310.KS	017810.KQ	097950.KQ
	Company	오뚜기	풀무원	CJ제일제당
	Ticker	Company		
	007310.KS	오뚜기	1.000000	0.288815
	017810.KQ	풀무원	0.288815	1.000000
	097950.KQ	CJ제일제당	0.312538	0.363255

## 샤프 비율

샤프지수는 위험 대비 수익률을 측정하는 투자 성과 지표

$$\text{Sharpe Ratio} = \frac{\text{포트폴리오 평균 수익률} - \text{무위험 수익률}}{\text{포트폴리오 위험}}$$

샤프 비율이 클수록 높은 수익을 적은 위험으로 얻는 이상적인 포트폴리오

## 샤프 비율 계산 함수

```
In [15]: # =====  
# 3. 샤프 비율 계산 함수 정의  
# =====  
def calculate_sharpe_ratio(weights, returns, risk_free_rate_daily):  
    port_ret = (returns @ weights).mean()  
    port_risk = (returns @ weights).std()  
    sharpe = (port_ret - risk_free_rate_daily) / port_risk  
    return port_ret, port_risk, sharpe
```

**weights:** 포트폴리오 내 각 자산의 투자 비중

**returns:** 각 자산의 일별 수익률

**risk\_free\_rate\_daily:** 일간 무위험 수익률



## 다양한 가중치 조합 계산

```
In [10]: # =====  
# 4. 다양한 가중치 조합 계산  
# =====  
weight_range = np.arange(0, 1.1, 0.1)  
risk_free_rate_daily = 0.03 / 252  
results = []  
  
for w1 in weight_range:  
    for w2 in weight_range:  
        w3 = 1 - w1 - w2  
        if 0 <= w3 <= 1:  
            weights = np.array([w1, w2, w3])  
            ret, risk, sharpe = calculate_sharpe_ratio(weights, returns, risk_free_rate_daily)  
            results.append((weights, ret, risk, sharpe))  
  
results_df = pd.DataFrame(results, columns=['Weights', 'Return', 'Risk', 'Sharpe'])  
results_df[['w1', 'w2', 'w3']] = pd.DataFrame(results_df['Weights'].tolist(), index=results_df.index)  
results_df = results_df.sort_values(by='Sharpe', ascending=False).reset_index(drop=True)  
results_df.head()
```

Out[10]:

	Weights	Return	Risk	Sharpe	w1	w2	w3
0	[0.2, 0.5, 0.30000000000000004]	0.002647	0.014998	0.168554	0.2	0.5	0.3
1	[0.2, 0.6000000000000001, 0.19999999999999996]	0.002824	0.016062	0.168411	0.2	0.6	0.2
2	[0.30000000000000004, 0.5, 0.19999999999999996]	0.002581	0.014630	0.168280	0.3	0.5	0.2
3	[0.1, 0.6000000000000001, 0.29999999999999993]	0.002890	0.016493	0.168010	0.1	0.6	0.3
4	[0.1, 0.7000000000000001, 0.19999999999999996]	0.003067	0.017664	0.166894	0.1	0.7	0.2

가장 샤프 비율이 높은 조합이  
Cj제일제당 0.2, 오투기 0.5, 풀무원 0.3으로 나타남  
그때의 샤프 비율은 0.168554

오투기의 가중치가 높을 때 샤프비율이 높게 나타남  
2024년 11월 ~ 2025년 5월까지 오투기가  
효과적 투자자산이었을 가능성이 큼

## 최적 포트폴리오 도출

```
In [11]: # =====  
# 5. 최적 결과 출력  
# =====  
best = results_df.iloc[0]  
print("\n 최적 포트폴리오")  
print(f"가중치: {best['Weights']}")  
print(f"평균 수익률: {best['Return']:.4%}")  
print(f"리스크 (표준편차): {best['Risk']:.4%}")  
print(f"샤프 비율: {best['Sharpe']:.4f}")
```

최적 포트폴리오  
가중치: [0.2 0.5 0.3]  
평균 수익률: 0.2647%  
리스크 (표준편차): 1.4998%  
샤프 비율: 0.1686

CJ제일제당 20%, 오투기 50%, 풀무원 30% 비중으로 투자했을 때

하루 평균 약 0.26%의 수익을 기대할 수 있음

일일 수익률의 변동 폭이 평균적으로  $\pm 1.5\%$  정도임

## 시각화 ① (리스크-수익률-샤프)

```
# =====  
# 6. 시각화  
# =====  
plt.figure(figsize=(8,6))  
sc = plt.scatter(results_df['Risk'], results_df['Return'], c=results_df['Sharpe'], cmap='viridis')  
plt.colorbar(sc, label='Sharpe Ratio')  
plt.xlabel('Risk (Std Dev)')  
plt.ylabel('Expected Return')  
plt.title('Portfolio Optimization Result')  
plt.grid(True)  
plt.show()
```

### 1. 위험-수익 관계

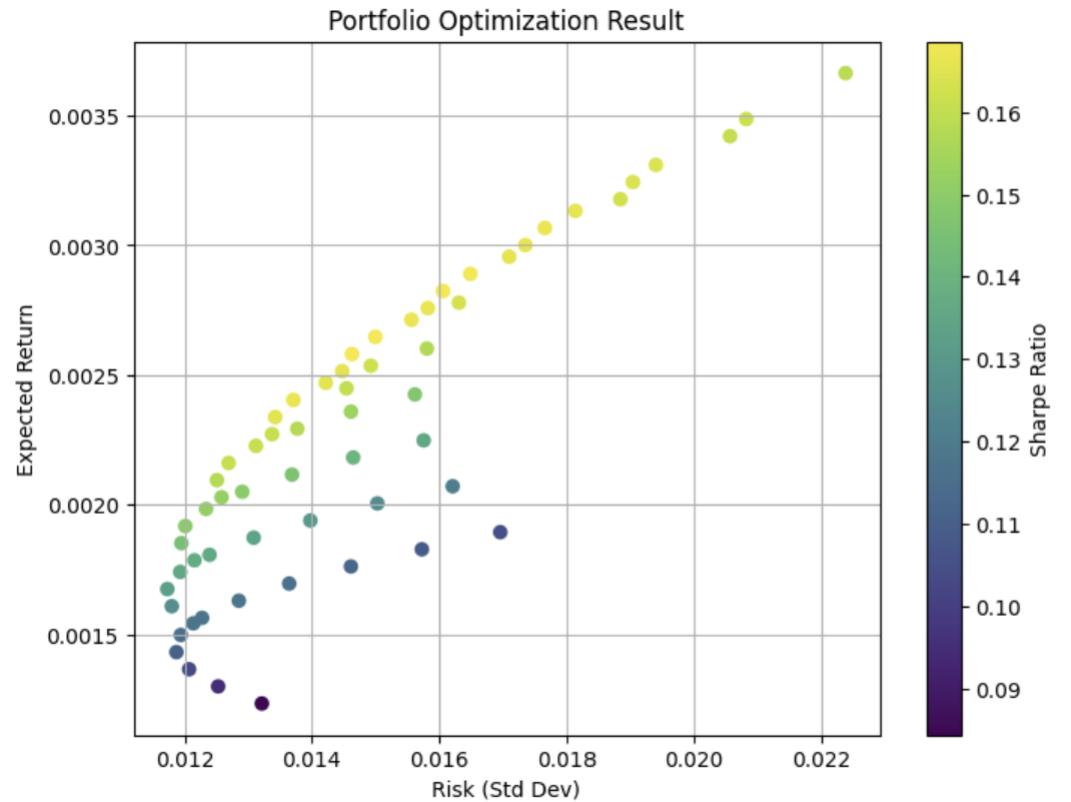
일반적으로 위험이 증가할수록 기대수익률도 증가하는 양의 상관관계

### 2. 최적 포트폴리오 식별

노란색 점들: 높은 샤프 비율을 가진 가장 효율적인 포트폴리오

### 3. 비효율적 구간

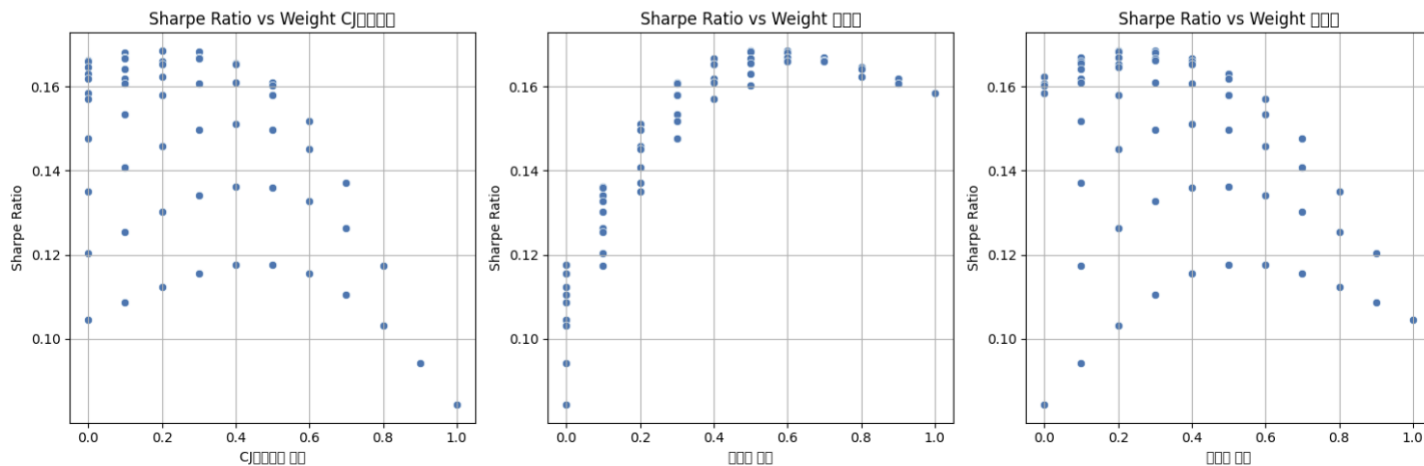
보라색/파란색 점들: 낮은 샤프 비율



## 시각화 ② (가중치 vs 샤프 비율)

```
# =====
# 7. 가중치 vs 샤프비율 시각화
# =====
plt.figure(figsize=(15, 5))
for i, w in enumerate(['w1', 'w2', 'w3'], 1):
    plt.subplot(1, 3, i)
    sns.scatterplot(data=results_df, x=w, y='Sharpe')
    plt.title(f'Sharpe Ratio vs Weight {names[i-1]}')
    plt.xlabel(f'{names[i-1]} 비중')
    plt.ylabel('Sharpe Ratio')
    plt.grid(True)

plt.tight_layout()
plt.show()
```



### 1. 최적 가중치 패턴

샤프 비율이 높은 구간에서의 각 자산 비중

C) 제일제당: 0.0~0.3 범위에서 높은 샤프 비율

오뚜기: 0.2~0.8 범위에서 높은 샤프 비율

풀무원: 0.1~0.6 범위에서 높은 샤프 비율

### 2. 자산별 기여도

오뚜기: 가장 넓은 비중 범위에서 높은 샤프 비율 유지

→ 포트폴리오의 핵심 자산

C) 제일제당: 낮은 비중에서 효과적 → 보조적 역할

풀무원: 중간 정도 비중에서 최적 → 균형 조절 역할

## RESULTS

### 1. 효율적 포트폴리오 구간 확인

위험도 0.0170.020, 기대수익률 0.00300.0035 구간에서 최적 샤프 비율 달성

### 2. 최적 자산 배분 전략

오프기: 40-60% (핵심 포트폴리오)

Cj제일제당: 0-30% (보조 역할)

풀무원: 10-60% (균형 조절)

### 3. 다양화 효과 입증

단일 자산 집중투자보다 적절한 분산투자가 위험 대비 수익률 크게 개선

극단적 비중(0% 또는 100%)은 샤프 비율 저하

두 번째 자산 중심의 균형 분산투자로 위험 대비 수익률 극대화가 가능하며,  
투자자의 위험 선호도에 따라 효율적 프론티어 구간 내에서 세부 조정 권장