7 주차

재무 데이터 처리 시 주의할 점

2023년 당기순이익 (-) ⇒ 적자기업
2022년 당기 순이익(-), 2023년 당기 순이익(-) ⇒ 적자 지속
2022년 당기 순이익(+), 2023년 당기 순이익(-) ⇒ 적자 전환
2022년 당기 순이익(-), 2023년 당기 순이익(+) ⇒ 흑자 전환

자사주(자기주식) 취득 및 소각의 의미

IF 소각, 발행주식 수 감소 = 주당 기업가치 상승

IF 취득, 유통 주식 수 감소 = 수급 호전(수급 경량화)

팩터

소형주 팩터

대형주에 비해 종목정보 부족, 재무 위험, 유동성 부족 → 비효율성
소형주 만의 체계적인 위험이 위험 프리미엄으로 반영되어 높은 수익률을 보인다.

- **위험 프리미엄**: 무위험 채권이 아닌 위험 자산을 보유하도록 유도하기 위해 위험 자산의 기대수익이 무위험 채권의 수익을 초과해야 하는 최소한의 금액

저위험 팩터

고변동성 주식 대비 상대적인 저평가(수급요인)
수익과 손실의 비대칭성
ㄴ> 변동성이 커지며 장기적으로 가격 가치가 점차 하락하는 현상

퀄리티 팩터

기본적 분석의 계량화

수익성, 안정성, 성장성 지표 등의 분위별 성과비교(ex 매출 총이익 기준)

모멘텀 팩터

한방향으로 지속적으로 변동하려는 경향 (관성, 대세를 따라간다)

고배당 팩터

배당할인 모형, 배당수익률(DY)이 높을 수록 주식 수익률이 높다는 의견

밸류 팩터

내재 가치 대비 저평가된 종목의 투자 성과가 좋다 (저PER, 저PBR 투자 성과 굿)

멀티 팩터

팩터간의 결합
낮은 상관관계 분산효과로 포트폴리오 안전성 증대

1) 혼합(Mix) : 서로 다른 팩터에 각각 투자

2) 통합(Integrate) : 각 종목의 팩터 순위를 계산하여 합산 (Rank_sum)

3) 순차(Sequential)필터링

: A팩터 기준 상위 n% 걸러내고 이후 B팩터 기준 상위 N% 걸러낸다. ~~~

가치 투자와 모멘텀 투자

포트폴리오

가치주 포트폴리오

전통적인 가치지표인 PER와 PBR이 낮은 종목 선정을 통해 포폴 구성

각 지표 PER와 PBR의 순위를 구하고 다시 두 지표 순위의 합산을 구한다.

```
value_rank = data_bind[['PER', 'PBR']].rank(axis = 0) #  열방향 순위
value_sum = value_rank.sum(axis = 1, skipna = False).rank() # 행방향으로 순위합으로 다시 순위
data_bind.loc[value_sum <= 10, ['종목코드', '종목명', 'PER', 'PBR']]  # 합산 순위 10위 이내 추출
```

모멘텀 포트폴리오

과거 수익률이 높았던 종목에 투자

```
ret_list = pd.DataFrame(data=(price_pivot.iloc[-1] / price_pivot.iloc[0]) - 1,
                        columns=['return'])
data_bind = ticker_list[['종목코드', '종목명']].merge(ret_list, how='left', on='종목코드')
```

```
momentum_rank = data_bind['return'].rank(axis=0, ascending=False)
data_bind[momentum_rank <= 10]
```

멀티 팩터 포트폴리오

팩터간의 상관관계가 높지 않다면 복수 팩터를 통합적으로 고려 시 분산 효과를 기대할 수 있다.

```
value_sum_all = value_rank_all.sum(axis=1, skipna=False).rank()
data_bind.loc[value_sum_all <= 20]
```

퀄리티(우량주)포트폴리오 (하)

ex.수익성 지표 조합으로 구성

```
quality_sum = quality_rank.sum(axis=1, skipna=False).rank()
quality_list.loc[quality_sum <= 20,
                 ['종목코드', '종목명', 'ROE', 'GPA', 'CFO']].round(4)
```

섹터 중립 포트폴리오

팩터 전략의 단점 중 하나가 선택된 종목들이 특정 섹터로 쏠리는 경우가 많음(ex IT)
특히 과거 수익률을 토대로 종목 선정 시 특정 섹터 호황기에 종목이 동조되어 쏠림 현상 심화

쏠림 현상 해결법

전종목에 대해 모멘텀을 순위화 하여 종목 추출하지 않고 먼저 섹터별로 구성종목들의 모멘텀을 정규화 시킨 후 다시 모든 섹터 종목들을 대상으로 정규화시킨 값을 순위화한다. → 특정 섹터 집중 현상 완화

포트폴리오 투자 성과 지표

샤프, MDD CAGR 만 보자(그정도는 알고 있자고 함)

1) 샤프지수(Sharp Ratio) = $\frac{\text{수익률-무위험수익률}}{\text{수익률의 표준편차}}$

Sortino 지수= $\frac{\text{수익률-무위험수익률}}{\text{(-)수익률의 표준편차}}$

5) MDD(Maximum Drawdown) =최대낙폭
특정 투자기간에서 포트폴리오의 고점에서 저점까지 최대 누적손실액

$$MDD = \frac{Target\ Value - Peak\ Value}{Peak\ Value}$$

6) CAGR(Compound Annual Growth Rate) : 기하평균을 이용한 연복리 수익률

$CAGR(t_0, t_n) = \left(\frac{V(t_n)}{V(t_0)}\right)^{\frac{1}{t_n - t_0}} - 1$, Return $= \frac{V(t_n)}{V(t_0)} - 1$, $t_n - t_0$= 투자년수, V는 시점별 투자가치

베타 값 구하기 (알아두기, 재무성과 백테스팅.ipynb에 있는 듯)

```
def beta(df, market=None):  # 베타값을  행렬계산으로 한방에 다 구하기

    if market is None:
        market = df['kospi']
        df = df.drop('kospi', axis=1) # df: 개별 종목수익률
    X = market.values.reshape(-1, 1)   # 재구조화 (-1) 행기준으로  X: kospi 수익률
    X = np.concatenate([np.ones_like(X), X], axis=1)
    b = np.linalg.pinv(X.T.dot(X)).dot(X.T).dot(df.values)
    return pd.Series(b[1], df.columns, name=df.index[-1])
```

```
# 전종목 일별 롤링 베타구하기
def roll(df, w):
    for i in range(df.shape[0] - w + 1):   # 롤링 기간 설정
            yield pd.DataFrame(df.values[i:i+w, :], df.index[i:i+w], df.columns)
## roll 함수 실행
betas = pd.concat([beta(adj_wr) for adj_wr in roll(adj_wr, 240)], axis=1).T22
```

```
# 종목컬럼에 해당 index에서 계산된 종목별 베타 추출  (squeeze는 차원축소 )
some_beta= betas[betas.index=='2018-01-02'].squeeze()
```

K-Ratio (누적 수익률의 기울기 / 표준 오차)

동일한 누적 수익률을 가진 두 종목이 있다고 가정해보자.

- A의 경우 상승 폭이 작다가 최근 급등으로 누적 수익률이 높아진 경우
- B의 경우 꾸준히 상승해 누적 수익률이 높아진 경우
- 우리는 꾸준하게 상승한 B가 더 매력적인 모멘텀 주식이라고 본다.
  모멘텀의 꾸준함을 측정하는 지표가 K-Ratio

- K-ratio 구하는 코드 있는데 종목 별로 회귀 분석을 실시한다.

```
ret = price_pivot.pct_change().iloc[1:]   # 종목별  일간 수익률 1행은 NA, 2행부터
ret_cum = np.log(1 + ret).cumsum()        # 종목별 누적수익률(로그사용)

#  날짜 길이 만큼 배열 생성 (0 ~ 63 ) :  x는  시간흐름을 의미
#  첫번째 종목의 누적수익률  :  Y는 종목 누적수익률을 미미
x = np.array(range(len(ret)))
y = ret_cum.iloc[:, 0].values

# 회귀 분석 실시
reg = sm.OLS(y, x).fit()

# 여기서 K-ratio는
k-ratio = (reg.params / reg.bse)이다.

## 종목 별 회귀분석 후 k-ratio 구하기
x = np.array(range(len(ret)))
k_ratio = {}

for i in range(0, len(ticker_list)):

    ticker = data_bind.loc[i, '종목코드']

    try:
        y = ret_cum.loc[:, price_pivot.columns == ticker]
        reg = sm.OLS(y, x).fit()
        res = float(reg.params / reg.bse)
    except:
        res = np.nan

    k_ratio[ticker] = res
```

ETF

특정지수 및 특정 자산의 가격 움직임과 수익률이 연동되도록 설계된 상장지수집합투자증권
거래소에 상장되어 주식처럼 거래되는 펀드

ETF 장점

소액으로 우량주 또는 특정 섹터에 분산 투자하는 효과
펀드보다 수수류가 싸다
매도시 거래세가 없다.

ETF 용어

- NAV(순자산가치): 편입하고 있는 주식, 현금, 배당, 이자 소득 등을 모두 더한 가치의 합, 순 자산을 ETF의 주식 수로 나눈 것이 순자산 가치
- 괴리율: (시장 가격 - NAV) / NAV * 100 - NAV와 현재 시장 가치와의 차이
- 추적오차: ETF가 궁극적으로 추종해야 하는 기초 지수를 따라가지 못하는 정도
- 보수율: ETF의 총 비용은 운용 수수료와 기타 비용의 합
  ETF의 운용 및 서비스의 대가로 투자자가 지불하는 비용의 비율

ETF 선택 조건

순 자산이 크고, 거래량이 많은 ETF(유동성 고려)
괴리율, 추적 오차, 보수율은 낮을수록 좋은 ETF이다.
