2. 금융 데이터 구조

- 4가지 형태
  - 기본데이터: 자산, 부채, 영업, 비용/수익, 거시변수
  - 시장데이터: 가격/이자율/내제 변동성, 거래량, 배당/쿠폰, 시중 금리 등
  - 분석: 분석가 추천, 신용 평가, 뉴스 감성, 이익 예측 등
  - 대체 데이터: 위성/CCTV, 구글 검색, 트위터 등
- 기본데이터(Fundamental)

  대개 분기별 회계자료 이며, 후행되어 발표되는 것이 자주 일어남.

  데이터가 backfill 되거거나 수정되는 경우도 있음. → 이를 위해 최초 발표값과 수정값들로 저장하는 경우도 있음.

  잘 정규화 되어 있지만, 빈도가 적다. 시중에 많이 알려져 유용성이 크지 않을 수 있지만, 다른 자료와 종합적으로 볼 때 가치를 가질 수 있음
- 시장데이터

  시장 상황을 전체를 재구성할 수 있는 데이터를 제공하는 업체도 있음. 모든 시장 참여자는 기록을 남기기 때문에, 경쟁자들의 다음 전략을 예측할 수 있음. 데이터가 방대하며, 전략연구에 흥미로운 데이터 셋.
- 분석

  기본, 시장, 대체 등의 원시 데이터로부터 파생된 데이터로 볼 수 있음.
- 대체 데이터

  private(SNS, 뉴스 웹 탐색), 비즈니스 프로세스(거래내역, 사내 데이터, 정부 기관), 센서(위성, 기상, CCTV) 등이며, 이들은 최초 정보라는 것이 차별된다.

  선박의 흐름, 주차장 여유공간 등은 선행될 수 있음.

- 바(bar)

  표준 바와 정보주도 바(information-driven bar)로 구분할 수 있다.
  - 표준바
    - 시간바

      가장 보편적이지만, 단점은 시장은 정보를 일정한 시간 간격으로 처리하지 않음.(ex. 시장 시작 1시간은 정오보다 훨씬 거래가 활발함) 또한, 시간에 따라 추출된 시계열 자료는 종종 좋지 않은 성질을 보임. (Ex. serial correlation, heteroscedasticity, 수익률의 비정규성)
    - 틱바

      사전에 정한 거래 건수가 발생할 때마다 추출하는 것. 고정된 거래 건수에 따른 가격 변동은 가우시안 분포를 따를 수 있지만, 고정된 기간의 가격 변동은 분산이 무한이 큰 paretian 분포를 따를 수 있음.

      거래활동에 관한 함수로 샘플링하면, IID 정규분포에 근접한 수익률을 얻을 수 있음이 여러 연구에서 발표됨. 따라서, 틱바는 시간바보다 보다 유용한 통계적 추론이 가능할 수도 있음.

      틱바 구성시, 이상치에 주의 해야 함. 특히나 장 시작이나 종료에는 주의해야할 수 있다.
    - 거래량바

      틱바는 주문의 파편성으로 틱 수가 임의적일 수 있음 → 이를 미리 정의된 단위의 거래(주, 선물 계약 등)이 일어 날 때마다 샘플링

      거래량에 기반한 수익률 샘플링이 틱바에 의한 샘플링에 비해 더 나은 통계적 성질(IID 등)을 가진다는 연구가 있음.

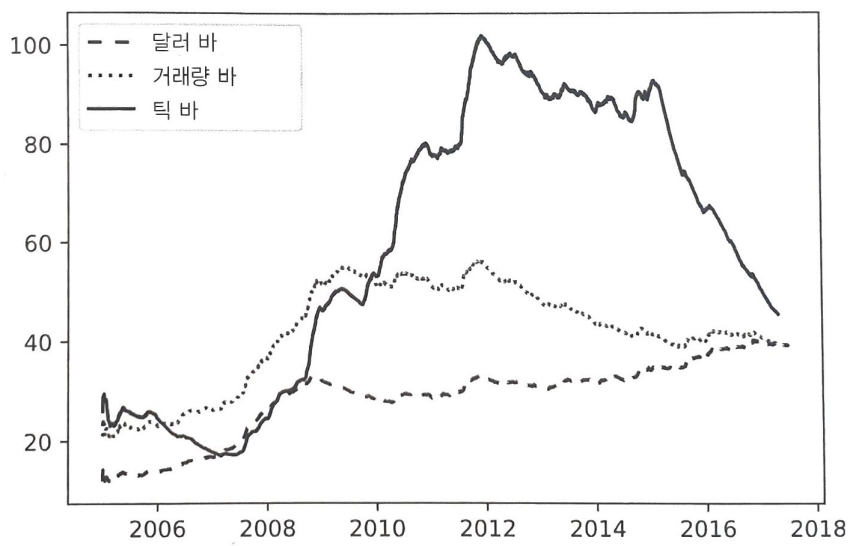
      또한, 시장 미시구조 이론이 가격과 거래량 사이의 상호작용을 연구 함.
    - 달러바

      사전에 정해둔 시장의 가치(ex 화폐)가 거래될 때마다 관측값을 샘플링 함.

      달러바를 사용하는 이유는 주식 수는 실제 거래된 가치의 함수이기 때문이다.

두 분째 이유는 총발행 주식 수가 종종 기업 행위의 결과로 주식이 거래되는 동안 여러번 변경되기 때문이다. 새 주식 발행이나 자사주 매입 등 다양한 기업행위에 있어 달러바는 강건한 경향이 있다.

아래 그림은 고정된 바크기로, 틱, 거래량, 달러 바를 표현한 그림의 예시이다.



- 정보주도 바(information-driven bar)

  시장에 새로운 정보가 도달한 경우 더 빈번히 샘플링하기 위함이며, 정보는 시장 미시 구조 측면에서 사용. 미시 구조 이론은 불균형한 부호의 거래량이 지속되는 데 주안점을 두고 있는데, 이 현상은 정보-기반 거래자의 존재 여부와 연계돼 있기 때문임.(19장에서 상세히)

  - 틱균형바(Tick Imbalnace Bar)

    기본아이디어: 틱의 불균형이 예상을 초과할 때마다 샘플링하자.

    틱 시퀀스: $\{(p_t, v_t)\}_{t=1,\dots,T}$

    가격: p, 거래량: v

    $$b_t = \begin{cases} b_{t-1}, & \text{if } \Delta p_t = 0 \\ \frac{|\Delta p_t|}{\Delta p_t}, & \text{if } \Delta p_t \neq 0 \end{cases}$$

    $$b_t \in \{-1, 1\}$$

    주어진 임계값을 넘는 틱 인덱스 T값을 결정하는 절차

    1) $\theta_T = \Sigma_{t=1}^{T} b_t$

    2) $E_0[\theta_T] = E_0[T](P[b_t = 1] - P[b_t = -1])$

    틱바의 크기: $E_0[T]$

    틱이 매수로 불류될 비조건부 확률: $P[b_t = 1]$

    틱이 매도로 분류될 비조건부 확률: $P[b_t = -1]$

    $P[b_t = 1] + P[b_t = -1] = 1$ 이므로

    $E_0[\theta_T] = E_0[T](2P[b_t = 1] - 1)$

    $E_0[T]$는 이전 바들로부터 T 값의 지수 가중 이동 평균으로 계산 가능.

    $2P[b_t = 1] - 1$ 는 이전 바들의 bt값의 지수 가중 이동평균으로 계산 가능

    3) TIB는 T^*-contigous의 부분집합으로 정의

    $$T^* = \arg\min_T \{|\theta_T| \geq E_0[T]|2P[b_t = 1] - 1|\}$$

    불균형의 크기: $|2P[b_t = 1] - 1|$

    θ_T가 예상보다 더 불균형인 경우 작은 T가 많이 조건을 만족(?)

    이러한 TIB는 정보 비대칭으로 인한 정보기반 거래자가 있는 경우 더 빈번히 발생하며 TIB를 동일한 정보를 가진 거래의 버킷으로 이해할 수 있다.

  - 거래량 불균형 바(VIB)/달러 불균형 바(DIB)

    VIB, DIB는 TIB의 개념을 확장

    이러한 변환은 기업 행위와 관련된 이슈(추가 발행 등)를 어느 정도 해소 할 수 있음.

  - 틱 런 바(TRB)

TIB, VIB, DIB는 주문 흐름의 불균형을 각각, 틱, 거래량, 거래된 금액 가치 기준으로 모니터링 함. 대규모 거래자들은 오더북을 전부 휩쓸어 가는 방안, 아이스버그 주문(대규모 주식을 거래하되 매 거래 시 미리 정한 소량으로 집행해 전체 거래량을 숨기는 거래 기법)을 하는 방안, 부모 주문을 자식 주문으로 쪼개는 등의 방법을 사용한다. 이런 경우 시퀀스안에 거래 흔적을 남기며, 거래량 안에서 매수 시퀀스를 조사하는 것이 유용하다.

매수 시퀀스가 기대에서 벗어날 경우 샘플링하는 방법이다.

- 거래량 런 바 / 달러 런 바

  거래량이나 달러 거래의 특정 방향의 거래가 기대값을 초과 한 경우 샘플링하는 것.

- 복수 상품 계열 다루기

  연구 중인 시계열의 속성을 변경하는 이벤트(배당, 기업행위, 쿠폰 등)에 의해 연구 중인 시계열의 속성을 변경하는 이벤트를 동적으로 적절히 모델링하고자 할 수 있음.

  이러한 경우, 연구 노력이 물거품 될 수 있는 구조적 변화가 일어날 수 있음(17장)

  증권의 바스켓을 단일 현금 상품처럼 모델링할 수 있는 ETF 트릭을 알

  복잡한 멀티 상품 데이터셋을 total-return ETF를 따르는 단일 데이터셋으로 변환하는 것이 목적이며, 이는 기초 자산의 복잡도나 구성에 상관 없이 코드를 통해 현금성 상품(만기가 없는 현금성 상품)만 거래하는 것으로 가정할 수 있기 때문에 유용하다.

  - ETF 트릭

    선물 스프레드 전략을 수립한다 가정. 이는 완전 상품과 아래 부분이 다름.

    선물 스프레드: 동일한 선물간 가격 괴리에서 얻는 거래

    a. 스프레드는 시간에 따라 변동하는 비중 벡터로 특징 지을 수 있음.

    b. 스프레드는 가격을 반영하지 않으므로 음의 값을 가질 수 있음

    c. 모든 구성요소의 거래시점이 일치하지 않아 스프레드는 가장 최근 가격 레벨에서 거래되는 것이 아니며 즉시 거래되는 것도 아님. 그리고 매수 매도 크로싱 등 거래 실행 비용도 항상 고려해야 함.

    ex. 스프레드에 1달러 가치를 투자한 경우

    $o_{i,t}, p_{i,t}, \varphi_{i,t}, \nu_{i,t}, d_{i,t}$ 는 각각 금융상품 i의 시가, 종가, 포인트당 달러가치, 거래량, 캐리/배당/쿠폰에 의한 가치. 바 t=1,….,T에서의 각 금융상품은 거래가 가능하나, [t-1, t]전체에 걸쳐서는 거래가 불가능하더라도 적어도 t-1, t에서는 거래가 가능.

    이 때, 선물 바스켓의 **1달러 투자 가치** $\{K_t\}$는 다음과 같이 유도 가능. $B \subseteq \{1, ..., T\}$

    $$h_{i,t} = \begin{cases} \frac{w_{i,t} K_t}{o_{i,t+1} \varphi_{i,t} \Sigma_{i=1}^{I} |w_{i,t}|} & if\ t \in B \\ h_{i,t-1} & otherwise \end{cases}$$

    $$\delta_{i,t} = \begin{cases} p_{i,t} - o_{i,t} & if\ (t-1) \in B \\ \Delta p_{i,t} & otherwise \end{cases}$$

    $$K_t = K_{t-1} + \Sigma_{i=1}^{I} h_{i,t-1} \varphi_{i,t} (\delta_{i,t} + d_{i,t})$$

    최초운용자산 AUM(Asset Under Management) $K_0 = 1$

    $h_{i,t}$: 시간 t에서 금융상품 i의 보유 자산(주식 수 또는 계약 수)

    $\delta_{i,t}$: 금융상품 i의 시점 t-1과 t사이의 시장 가격 변동. $t \in B$일 때마다 수익이나 손실은 재투자되어 음수가격 방지

    $d_{i,t}$: 배당금

    $h_{i,t}$에서 w(오메가)부분의 목적은 레버리지를 낮추기 위함. (w는 배분벡터, 다음 절에 설명)

    τ_i를 금융상품 i의 1달러당 거래 비용이라 가정. ex. $\tau_i = 1e-4$

    이 때, 모든 관측 바 t에 대해 전략에 필요한 추가 변수

    - 재조정비용(rebalance cost): 배분 재조정에 연계된 변동비용, 배분을 재조정할 때 스프레드 매도가 허구의 이익 발생시킴. 이를 차감해야함.

      $$c_t = \sum_{i=1}^{I} (|h_{i,t-1}| p_{i,t} + |h_{i,t}| o_{i,t+1}) \varphi_{i,t} \tau_i, \forall t \in B.$$

    - 매수 매도 호가 차이 (bid-ask spread): 가상의 ETF 한 단위를 매수하거나 매도하는 비용.

      $$\tilde{c}_t = \sum_{i=1}^{I} |h_{i,t-1}| p_{i,t} \varphi_{i,t} \tau_i$$

- 거래량(volume): 바스켓상의 가장 거래가 안된 상품에 의해 결정

$$v_t = \min_i \left\{ \frac{v_{i,t}}{|h_{i,t-1}|} \right\}.$$

거래비용함수는 반드시 선형일 필요가 없으며, 비선형 비용함수를 위 정보에 기초해 시뮬레이션 할 수 있음. ETF 트릭을 통해 선물의 바스켓을 만기가 없는 단일 현금성 상품과 같이 모델링 가능.

- PCA Weights

ETF 트릭에서 배분 벡터 w를 도출하는 방법 중 하나.

N개의 금융상품에서 Nx1 크기의 평균이 μ인 벡터에와 NxN의 공분산행렬 V인 IID 다변량 가우시안 과정에서

a. 스펙트럼분해(정규 또는 대칭행렬에 대한 고유값분해)를 수행. $VW = W\Lambda$ 고유값행렬이 내림차순이 되도록 정렬

b. 벡터 정렬 w가 주어질 때 포트폴리오 리스크 정의

$$\sigma^2 = \omega' V \omega = \omega' W \Lambda W' \omega = \beta' \Lambda \beta = (\Lambda^{1/2}\beta)'(\Lambda^{1/2}\beta)$$

c. Λ는 대각행렬이기 때문에 $\sigma^2 = \Sigma_{n=2}^{N} \beta_n^2 \Lambda_{n,n}$이고, n번째 성분에 해당하는 리스크는 아래와 같다. 그리고 $\{R_n\}_{n=1,\ldots N}$은 직교 성분에 따른 리스크 분포로 해석할 수 있다.

$$R_n = \beta_n^2 \Lambda_{n,n} \sigma^{-2} = [W'\omega]_n^2 \Lambda_{n,n} \sigma^{-2}$$

d. 사용자 정의 리스크 분포 R에 대하여 β를 표현하면

$$\beta = \left\{ \sigma \sqrt{\frac{R_n}{\Lambda_{n,n}}} \right\}_{n=1,\ldots,N}$$

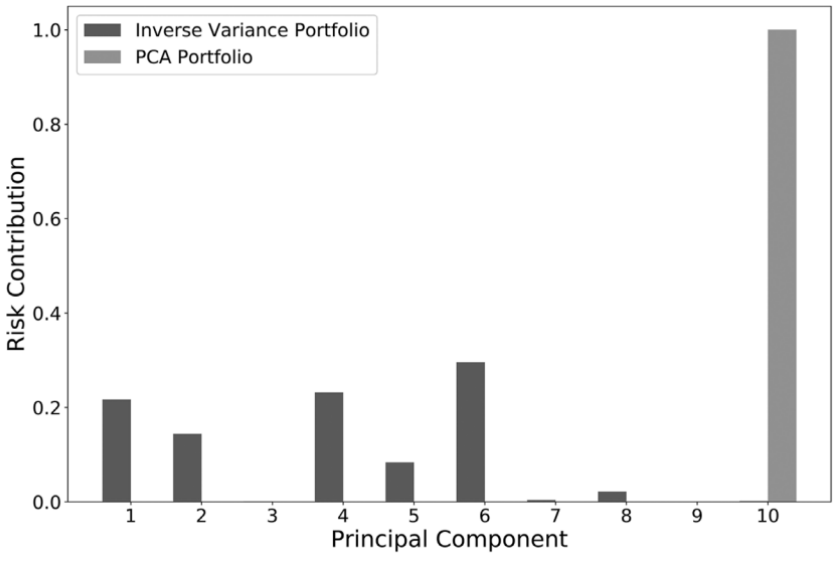
e. 배분 벡터 w 계산. $w = W\beta$

위 과정을 코드로 표현하면 아래와 같다.

```
def pcaWeights(cov: np.ndarray, riskDist: np.ndarray = None,
               riskTarget: float = 1.) -> np.ndarray:
    eVal, eVec = np.linalg.eigh(cov)
    indices = eVal.argsort()[::-1]
    eVal, eVec = eVal[indices], eVec[:, indices]
    if riskDist is None:
        riskDist = np.zeros(cov.shape[0])
        riskdist[-1] = 1.
    loads = riskTarget * (riskDist / eVal) ** 0.5
    weights = np.dot(eVec, np.reshape(loads, (-1, 1)))
    return weights
```

아래 그림은 역분산배분에 따른 주성분별 리스크 기여이다. 최대 분산(주성분 1,2)를 포함한 대부분의 주성분이 리스크에 영향을 미치며, 반대로 PCA 포트폴리오는 최소 분산만 리스크에 영향을 미친다.

코드에서 사용자 지정 리스크 분포 R이 ristDist인자로 전달되며 None인 경우 최소 고유값의 주성분에 배분되는 것을 가중하며, weight는 σ(riskTarget)와 일치하도록 재조정된다.



- 단일 선물 롤오버

  ETF트릭을 사용하면 단일 선물 계약의 롤(만기연장)은 1-leg(스프레드의 한쪽 면) 스프레드의 특별한 경우로 간주해 다룰 수 있다. 보다 직접적인 방법으로는 누적 롤 갭 시계열을 형성한 후 그 갭 시계열 만큼을 가격 시계열에서 차감하는 것.

  FUT_CUR_GEN_TICKER: 가격에 연계된 계약 식별, 각 롤오버마다 변함

  PX_OPEN, PX_LAST: 바와 연계된 시가, 종가

  VWAP: 바와 연계된 거래량-가중평균 가격

```
def getRolledSeries(pathIn,key):
  series=pd.read_hdf(pathIn,key='bars/ES_10k')
  series['Time']=pd.to_datetime(series['Time'],format='%Y%m%d%H%M%S%f')
  series=series.set_index('Time')
  gaps=rollGaps(series)
  for fld in ['Close','VWAP']:
    series[fld]-=gaps
  return series
#-------------------------------------------
def rollGaps(
  series,
  dictio={'Instrument':'FUT_CUR_GEN_TICKER',
          'Open':'PX_OPEN',
          'Close':'PX_LAST'},
  matchEnd=True
):
  # Compute gaps at each roll, between previous close and next open
  rollDates=series[dictio['Instrument']]\
            .drop_duplicates(keep='first').index
  gaps=series[dictio['Close']]*0
  iloc=list(series.index)
  iloc=[iloc.index(i)-1 for i in rollDates] # index of days prior to roll
  gaps.loc[rollDates[1:]]=series[dictio['Open']].loc[rollDates[1:]]- \
                          series[dictio['Close']].iloc[iloc[1:]].values
  gaps=gaps.cumsum()
  if matchEnd:
    gaps-=gaps.iloc[-1] # roll backward
  return gaps
```

  일반적으로 음이 아닌 롤 시계열과 작업하고 싶을 때, 다음 처럼 투자 1달러당 가격 시계열을 도출 가능하다.

```
raw=pd.read_csv(filePath,index_col=0,parse_dates=True)
gaps=rollGaps(raw,
    dictio={'Instrument':'Symbol','Open':'Open','Close':'Close'})
rolled=raw.copy(deep=True)
for fld in ['Open','Close']:
  rolled[fld]-=gaps
rolled['Returns']=rolled['Close'].diff()/raw['Close'].shift(1)
rolled['rPrices']=(1+rolled['Returns']).cumprod()
```

- 특성샘플 추출

  지금까지 비정형 금융 데이터 집합으로 연속이고 동질이며 구조화된 데이터를 생성하는 방법을 배움. ML을 통해 이러한 적용해 보려 시도할 수 있지만, 표본이 큰 경우 효율적이지 않거나, 분류문제로 변환하는 경우 더 잘 작동하는 경우가 있다.

  - 축소를 위한 샘플링

    아래와 같이 표본을 줄이기 위한 단순한 다운샘플링은 예측력 관점에서 연관성을 갖는 관측을 찾기 어렵다는 단점이 있음.

    linspace sampling: 일정한 크기로 순차적 표본 추출

    uniform sampling: 균등분포를 사용한 무작위 표본 추출

  - 이벤트 기반의 샘플링

    포트폴리오 매니저는 구조적변화, 추출된 신호, 미시 구조적 현상 등의 어떤 사건의 발생 후에 배팅. 이러한 이벤트는 변동성 확대, 균현 레벨에서 스프레드의 이탈, 거시경제 통계량 발표 등과 연계되어 있을 수 있음.

    이벤트 기반의 표본 추출 기법의 예시

    - CUMSUM 필터

      CUMSUM 필터를 통해 품질을 통제하여 측정값이 목표값의 평균과 얼마나 벗어나는지 확인해보자.

      locally stationary process의 IID 관측값 $\{y_t\}_{t=1,\dots,T}$ 의 누적합의 정의는 아래와 같다.

$$S_t = \max\left\{0, S_{t-1} + y_t - \mathrm{E}_{t-1}\left[y_t\right]\right\}$$

$S_0 = 0$ 이며, 임계값 h에 대해 $S_t \geq h$인 경우 이탈로 볼 수 있을 것이다. 임계값에 대하여 아래와 같을 때 필터는 탐지한다.

$$S_t \geq h \Leftrightarrow \exists \tau \in [1, t] \left| \sum_{i=\tau}^{t} \left(y_i - \mathrm{E}_{i-1}\left[y_t\right]\right) \geq h \right.$$

그리고 상방 이탈인지, 하방 이탈인지에 대해 아래와 같이 필터를 구분할 수 있다.

$$S_t^+ = \max\left\{0, S_{t-1}^+ + y_t - \mathrm{E}_{t-1}\left[y_t\right]\right\},\ S_0^+ = 0$$

$$S_t^- = \min\left\{0, S_{t-1}^- + y_t - \mathrm{E}_{t-1}\left[y_t\right]\right\},\ S_0^- = 0$$

$$S_t = \max\left\{S_t^+, -S_t^-\right\}$$

lam and Yam, 1997에서는 h가 이전 고점이나 저점에서 발견될 때 매수 매도 신호를 하는 투자전략을 제안했으며, fama an bume, 1966에서의 필터거래 전략도 이와 같다는 것을 증명.

이를 코드화하면 아래와 같고, $E_{t-1}[y_t] = y_{t-1}$ 이다.

```
def getTEvents(gRaw,h):
  tEvents,sPos,sNeg=[],0,0
  diff=gRaw.diff()
  for i in diff.index[1:]:
    sPos,sNeg=max(0,sPos+diff.loc[i]),min(0,sNeg+diff.loc[i])
    if sNeg<-h:
      sNeg=0;tEvents.append(i)
    elif sPos>h:
      sPos=0;tEvents.append(i)
  return pd.DatetimeIndex(tEvents)
```

cumsum 필터가 실용적이고 매력적인 이유 중 하나는 임계값 수준에서 변동하는 원시계열(gRaw)에서 볼린저 밴드(이평선과 이평선의 표준편차 기반)같이 여러 매수매도 신호가 나오는 것에 강인하다는 것.

변수 S_t는 구조적 변화 통계량, 엔트로피, 시장 미시척도 등에서 17~19장에서 더 깊게 다룬다. ex. SADF(supremum augmented dickey-fuller) 기반의 이벤트 샘플링 등이 있다.

이벤트 탐지를 통해 이벤트 기반의 바를 구성해 ML을 적용할 수 있음