Chapter 2
Financial Data Structures

YONSEI UNIVERSITY

CONTENTS

01.

Motivation

1 Motivation

- **목적**
  - 금융 데이터 분류 이해
  - 비정형 데이터 정형화 방법 습득
    - 원시 데이터에서 의미 있는 특징 추출
- **강조점**
  - 사람이 만든 데이터에는 오류가 포함될 가능성이 다분함
  - 데이터를 다루기 위해서는 해당 오류를 식별할 수 있는 식견을 갖춰야 함
- **중요 지표**
  - 좋은 통계적 특징
  - 계산 용이성

02.

Essential Types of Financial data

2 Essential Types of Financial data

	Fundamental Data	Market Data	Analytics	Alternative Data
종류	• Assets • Liabilities • Sales • Costs/Earnings • Macro Variables	• Price/Yield/Implied Volatility • Volume • Dividend/Coupons • Open Interest • Quotes/Cancellations • Aggressor Side	• Analyst Recommendations • Credit Rating • Earnings Expectations • News Sentiment	• Satellite/CCTV Images • Google Searches\ • Twitter/chats • Meta data
특성	• 대부분 회계 데이터 • 분기별 수집 • 저빈도 데이터	• 거래는 정보의 흔적을 남김 • 데이터 수 많음	• 필요한 정보들이 이미 추출	• 근본 정보 • 독창적인 정보 생성 가능

유의점	• 조사 시점과 발표 시점 사이의 오차
	  • 많은 연구들의 오류 내포 원인
	• 미래 데이터 사용
	  • 과거 자료 발간 후 수정 거침
	  • 미래 정보가 들어가 있는 경우가 많음
		• 방대한 데이터 처리에 어려움	• 편향 가능성
			• 타자들도 해당 정보 인지
				• 처리에 많은 비용
				• 개인정보문제
				• 저장 및 관리에 어려움

활용 방안	• 다른 데이터와의 통합	• TWAP 알고리즘을 이용한 Front Run • GUI 투자자들의 단위 숫자 지지 및 저항		

03.

Bars

3 Bars

- **Bars**
  - **정의**
    - 테이블 : 정보가 추출되어 테이블 형태로 표현된 데이터
    - Bars : 해당 테이블의 각 행
  - **특징**
    - 대부분의 머신러닝 알고리즘의 입력 형식
  - **분류**
    - Standard Bars
    - Information-Driven Bars

3 Bars

- **Standard Bars**
  - **정의**
    - 비균질 계열 데이터를 균질 계열화한 Bars
  - **특징**
    - 쉽게 구할 수 있음(API 등)
  - **종류**
    - Time Bars
    - Tick Bars
    - Volume Bars
    - Dollar Bars

3 Bars

- Standard Bars
  - Time Bars
    - **정의**
      - 설정한 **시간** 단위 샘플링
    - **특징**
      - 가장 보편적임
    - **유의점**
      - 오버 샘플링 및 언더 샘플링 가능성
        - 시장애서 거래는 시간에 따라 동질적이지 않음(장 개장 직후, 장 마감 전)
      - 좋지 않은 통계적 성질 보유
        - 계열 상관성, 이분산성, 수익률이 비정규분포성 등의 문제 발생

시각	open	high	low	close	volume	value
2020-01-28 09:00:00	59400.0	59400.0	59100.0	59400.0	1353089	80346295600
2020-01-28 09:00:30	59300.0	59400.0	58900.0	59100.0	268273	15853937600
2020-01-28 09:01:00	59000.0	59100.0	59000.0	59000.0	218328	12892250600
2020-01-28 09:01:30	59000.0	59200.0	59000.0	59200.0	197573	11677529000
2020-01-28 09:02:00	59100.0	59300.0	59100.0	59200.0	164103	9715396200
...	...	...	...	...	...	...
2020-01-28 15:28:00	59000.0	59000.0	58900.0	58900.0	0	0
2020-01-28 15:28:30	59000.0	59000.0	58900.0	58900.0	0	0
2020-01-28 15:29:00	59000.0	59000.0	58900.0	58900.0	0	0
2020-01-28 15:29:30	59000.0	59000.0	58900.0	58900.0	0	0
2020-01-28 15:30:00	58800.0	58800.0	58800.0	58800.0	1661807	97714251600

3 Bars

- **Standard Bars**
  - Tick Bars
    - **정의**
      - 설정한 **거래 건수** 단위 샘플링
    - **특징**
      - Time Bars보다 나은 통계적 성질
        - IID 정규분포에 근접한 수익률
        - *On the distribution of stock price differences(Mandelbrot, 1967)*
    - **유의점**
      - 이상치
        - 동시호가

시각	open	high	low	close	volume	value
2020-01-28 09:00:15	59400	59400	59300	59300	1139414	67681163000
2020-01-28 09:00:15	59300	59400	59300	59400	25205	1494664500
2020-01-28 09:00:15	59400	59400	59300	59400	259	15384500
2020-01-28 09:00:16	59400	59400	59300	59400	515	30544400
2020-01-28 09:00:16	59400	59400	59200	59300	12047	714386600
...	...	...	...	...	...	...
2020-01-28 15:19:50	58900	59000	58900	59000	56	3300800
2020-01-28 15:19:52	58900	59000	58900	59000	407	23997400
2020-01-28 15:19:53	58900	59000	58900	59000	1029	60618100
2020-01-28 15:19:55	58900	59000	58900	58900	176	10375700
2020-01-28 15:19:57	59000	59000	58800	58800	1671450	98282276200

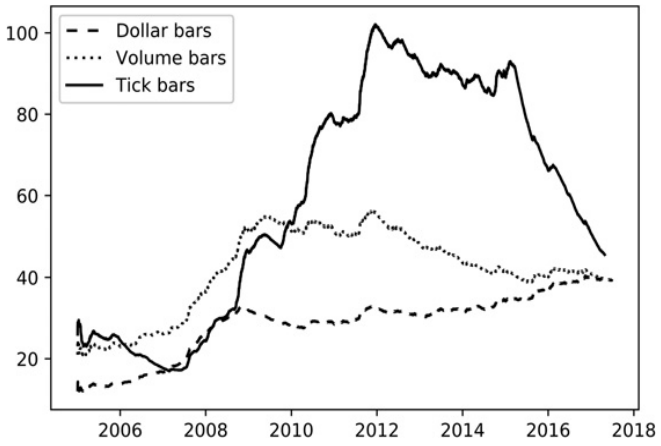
3 Bars

- **Standard Bars**
  - **Volume Bars**
    - **정의**
      - 설정한 **거래량** 단위 샘플링
    - **특징**
      - Tick Bars은 같은 거래량의 거래도 분할 횟수에 따라 다르게 처리
        - Ex) 주문량 10일 경우, 10개 1회 거래 V 1개 10회 거래
      - Tick Bars보다 더 좋은 통계적 성질
        - IID 정규분포에 근접한 수익률
        - *A subordinated stochastic process model with finite variance for speculative price(Clark, 1973)*

시각	open	high	low	close	volume	value
2020-01-28 09:00:15	59400	59400	59300	59300	1139518	67687330500
2020-01-28 09:00:15	59300	59400	59200	59200	59835	3547486800
2020-01-28 09:00:19	59300	59300	59200	59300	50018	2965372200
2020-01-28 09:00:24	59300	59400	59300	59300	35460	2103607200
2020-01-28 09:00:26	59200	59400	59100	59300	56480	3344029200
...	...	...	...	...	...	...
2020-01-28 15:15:31	58900	59000	58900	58900	48566	2862065700
2020-01-28 15:17:10	58900	59000	58900	58900	51789	3051350800
2020-01-28 15:18:00	58900	59000	58900	59000	49812	2935623200
2020-01-28 15:19:20	58900	59000	58900	58900	45162	2660847100
2020-01-28 15:30:29	58800	58800	58800	58800	1661807	97714251600

3 Bars

- **Standard Bars**
  - **Dollar Bars**
    - **정의**
      - 설정한 **거래대금** 단위 샘플링
    - **특징**
      - 가격 변동이 심할 경우 이전 기준들보다 동질적인 정보 제공
      - 액면분할, 병학 등의 주식 조정시에도 틱, 거래량에 비해 강건함



시각	open	high	low	close	volume	value
2020-01-28 09:00:15	59400	59400	59200	59300	1178483	69997927400
2020-01-28 09:00:16	59300	59400	59200	59400	81043	4804461900
2020-01-28 09:00:25	59400	59400	59100	59300	81785	4845436600
2020-01-28 09:00:28	59300	59400	59100	59100	89185	5281713600
2020-01-28 09:00:45	59200	59200	59000	59000	84733	5009011800
...	...	...	...	...	...	...
2020-01-28 15:12:04	58900	59000	58900	59000	84409	4976597200
2020-01-28 15:14:10	59000	59000	58900	58900	84762	4995529200
2020-01-28 15:16:45	58900	59000	58900	58900	84114	4956666000
2020-01-28 15:18:31	58900	59000	58900	58900	73206	4313463600
2020-01-28 15:30:29	58800	58800	58800	58800	1661807	97714251600

3 Bars

- ## Information-Driven Bars
  - **정의**
    - 시장에 새로운 정보가 유입되었을 때 더 많은 bar 생성
  - **용어**
    - **정보**
      - 거래량의 불균형이 정보를 반영한다는 가정
  - **종류**
    - Tick *Imbalance* Bars
    - **Volume/Dollar** *Imbalance* Bars
    - Tick *Runs* Bars
    - **Volume/Dollar** *Runs* Bars
  - **틱 규칙**
    - $b_t = \begin{cases} b_{t-1} & if\ \Delta p_t \\ \frac{|\Delta P_t|}{\Delta p_t} & if\ p_t \neq 0 \end{cases}$ where $b_t \in \{-1, 1\}$
    - 시장가 매수주문시 1, 시장가 매도 주문시 -1, 변동 없을 경우 이전 bar 따름

3 Bars

- ## Tick Imbalance Bars
  - **정의**
    - **틱 불균형**이 **예상**을 **초과**할 때마다 표본 추출
  - 1. $\theta_T = \sum b_t$
    - 시간 T에서의 틱 불균형
  - 2. $E_0[\theta_T] = E_0[T](P[b_t = 1] - P[b_t = -1])$
    - $E_0[\theta_T]$
      - Bar 시작점에서의 기댓값
    - $P[b_t = 1] + P[b_t = -1] = 1$
      - 틱이 매수로 분류될 확률 + 매도로 분류될 확률
    - $E_0[\theta_T] = E_0[T](2P[b_t = 1] - 1)$
      - $E_0[T]$: 이전 Bar에서 T 값의 지수 가중 이동평균(EWMA)
      - $2P[b_t = 1] - 1$: 이전 Bar b_t값의 지수 가중 이동평균(EWMA)
  - 3. TIB
    - $T^* = \arg\min\{|\theta_T| \geq E_0[T]|2P[b_t = 1] - 1\}$
      - $|2P[b_t = 1] - 1|$: 기대 불균형의 크기

3 Bars

- ## Volume/Dollar Imbalance Bars
  - **정의**
    - **거래량 / 거래대금 불균형**이 **예상**을 **초과**할 때마다 표본 추출
  - 1. $\theta_T = \sum b_t v_t$
    - v_t : 거래량 V 거래 대금
  - 2. . $E_0[\theta_T] = E_0\left[\sum_{t|b_t=1}^{T} v_t\right] - E_0\left[\left[\sum_{t|b_t=-1}^{T} v_t\right]\right] = E_0[T](P[b_t = 1]E_0[v_t|b_t = 1] - P[b_t = -1]E_0[v_t|b_t = -1])$
    - **치환**
      - $v^+ = P[b_t = 1]E_0[v_t|b_t = 1]$
      - $v^- = P[b_t = -1]E_0[v_t|b_t = -1]$
    - $E_0[T]^{-1}E_0[\sum_t v_t] = E_0[v_t] = v^+ + v^-$
    - $E_0[\theta_T] = E_0[T](v^+ - v^-) = E_0[T](2v^+ - E_0[v_t])$
      - $E_0[T]$: 이전 bar에서 $b_t v_t$값의 EWMA
  - 3. VIB / DIB
    - $T^* = \arg\min\{|\theta_T| \geq E_0[T](2v^+ - E_0[v_t])\}$
      - $(2v^+ - E_0[v_t])$: 기대 불균형의 크기

3 Bars

- Tick Runs Bars
  - **정의**
    - **전체 거래량** 대비 **매수 주문**이 **기댓값 벗어날 때** 표본추출
  - 1. $\theta_T = \max(\sum_{t|b_t=1}^{T} b_t \ - \sum_{t|b_t=-1}^{T} b_t)$
    - 현재 런의 길이
    - $b_t = 1, b_t = -1$ 따로 누적하여 두 수치 사이의 비대칭 기준
  - 2. . $E_0[\theta_T] = E_0[T]\max\{P(b_t = 1), 1 - P(b_t = 1)\}$
    - $E_0[\theta_T]$
      - Bar 시작시 기댓값
    - $E_0[T]$
      - 이전 Bar의 T값 EWMA
  - 3. TRB
    - $T^* = \arg\min\{|\theta_T| \geq E_0[T]\max\{P(b_t = 1), 1 - P(b_t = 1)\}\}$
      - 런의 기대 틱 횟수: $\max\{P(b_t = 1), 1 - P(b_t = 1)\}$
      - 시퀀스 단절 허용
        - 가장 긴 시퀀스 측정 대신 각 방향의 틱 개수 측정

3 Bars

- ## Volume / Dollar Runs Bars
  - **정의**
    - **거래량/달러 거래 방향이 기댓값 초과시** 표본 추출
  - 1. $\theta_T = \max(\sum_{t|b_t=1}^{T} b_t v_t - \sum_{t|b_t=-1}^{T} b_t v_t)$
    - 현재 런의 길이
  - 2. . $E_0[\theta_T] = E_0[T]\max\{P(b_t = 1)E_0[v_t|b_t = 1], (1 - P(b_t = 1))E_0[v_t|b_t = -1]\}$
    - $E_0[\theta_T]$
      - Bar 시작시 기댓값
    - $E_0[T]$: 이전 Bar의 T값 EWMA
    - $E_0[v_t|b_t = 1]$: 이전 Bar의 매수 거래량 EWMA
  - **3. TRB**
    - $T^* = \arg\min\{|\theta_T| \geq E_0[T]\max\{P(b_t = 1)E_0[v_t|b_t = 1], (1 - P(b_t = 1))E_0[v_t|b_t = -1]\}\}$
      - 런의 기대 거래량: $P(b_t = 1)E_0[v_t|b_t = 1], (1 - P(b_t = 1))E_0[v_t|b_t = -1]$
      - 시퀀스 단절 허용
        - 가장 긴 시퀀스 측정 대신 각 방향의 틱 개수 측정

04.

Dealing With Multi-Product Series

4 Dealing With Multi-Product Series

- **목적**
  - 동적으로 **대상의 가치가 변할 경우**의 이벤트 핸들링
    - 동적 가중치 조정이 필요한 금융 상품 모델링
    - 쿠폰, 배당 지급하는 상품 혹은 회사
      - Ex : 배당 재투자, 인덱스 구성 변경, 만기 롤오버
  - **ETF 트릭**
    - 증권의 바스켓을 단일 현금 상품인 것처럼 모델링 하는 방법
    - 복잡한 상품 데이터도 ETF 데이터로 변환 가능
      - 현금성 상품으로 거래하는 것처럼 인지 가능

4 Dealing With Multi-Product Series

- **ETF 트릭**
  - 선물 거래에 대한 이해
    - 완전한 상품이 아니라 스프레드를 다룸
      - **스프레드** : 시간에 대한 가중값 벡터
        - 가격이 수렴하지 않아도 스프레드 자체 수렴 가능
        - 이 경우 모델은 손익이 가중치의 수렴에서 기인한 것으로 오해 가능
      - 스프레드는 가격을 반영하지 않으므로 음의 값 가능
        - Non-negativity 조건
      - 거래 횟수가 모든 구성 요소에서 정확히 일치하지 않음
      - 가격 다이버전스 교차 같은 실행 비용 고려

4 Dealing With Multi-Product Series

- **ETF 트릭**
  - **해결 방법**
    - 스프레드에 투자된 1달러당 가치를 반영한 시계열 생성
      - **특징**
        - 양수
        - 거래비용 고려
    - ETF인 것처럼 모델링, 신호 생성, 거래

4 Dealing With Multi-Product Series

- **ETF 트릭**

- $h_{ij} = \begin{cases} \frac{w_{i,t}K_t}{o_{i,t+1}\varphi_{i,t}\sum|w_{i,t}|} & if\ t \in B \\ h_{i,t-1} & if\ t \notin B \end{cases}$
  - **정의**
    - 시간 t에서 금융상품 i의 보유 자산
  - $\frac{w_{i,t}}{\sum|w_{i,t}|}$: 배분에 있어 레버리지 낮추는 효과
    - $p_{i,t}$ 에 대해 잘 모를 수 있으므로$o_{i,t+1}$사용
- $\delta_{ij} = \begin{cases} p_{i,t} - o_{i,t} & if\ (t-1) \in B \\ \Delta p_{i,t} & if (t-1) \notin B \end{cases}$
  - **정의**
    - 금융상품 i에 대해 시간 t - 1과 t 사이의 시장 가격 변동
  - $\delta_{ij} \in B$일 경우 수익 및 손실 재투자
    - 음수 가격 방지
- $K_t = K_{t-1} + \sum h_{i,t-1}\,\varphi_{i,t}(\delta_{i,t} + d_{i,t})$
  - $K_0 = 1$

기표	기의
$o_{i,t}$	금융상품의 원 시초가
$p_{i,t}$	금융상품의 원 종가
$\varphi_{i,t}$	금융 상품 포인트당 USD 가치
$v_{i,t}$	금융 상품의 거래량
$d_{i,t}$	금융 상품 i의 바 t의서의 보유가치, 배당 또는 쿠폰 가치
$w_{i,t}$	배분 벡터
τ_i	거래 비용
B_t	t 시점의 바
$\boldsymbol{K_t}$	**배분벡터 $w_{i,t}$에 의해 특징지어진 선물 바스켓의 1달러에 대한 투자가치**

4 Dealing With Multi-Product Series

- ETF 트릭
  - **거래비용 관련 추가 고려사항**
    - **재분배 비용**
      - $c_t = \sum(|h_{i,t-1}|p_{i,t} + |h_{i,t}|o_{i,t+1})\varphi_{i,t}\tau\_i$
    - **매매 가격 차이**
      - $\bar{c} = \sum|h_{i,t-1}|p_{i,t}\varphi_{i,t}\tau_i$
    - **거래량**
      - $v_t = \min(\frac{v_{i,t}}{|h_{i,t-1}|})$
      - 바스켓 내의 최저 거래량

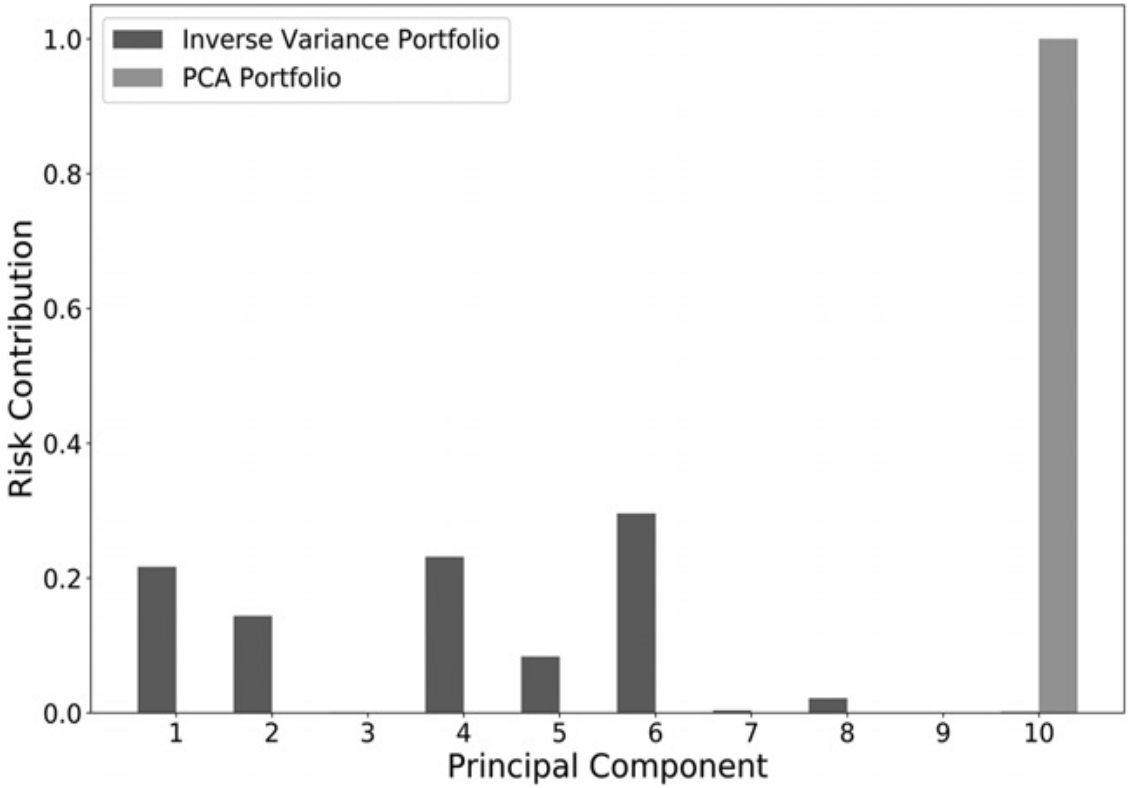
기표	기의
$o_{i,t}$	금융상품의 원 시초가
$p_{i,t}$	금융상품의 원 종가
$\varphi_{i,t}$	금융 상품 포인트당 USD 가치
$v_{i,t}$	금융 상품의 거래량
$d_{i,t}$	금융 상품 i의 바 t의서의 보유가치, 배당 또는 쿠폰 가치
$w_{i,t}$	배분 벡터
τ_i	거래 비용
B_t	t 시점의 바
K_t	배분 벡터 $w_{i,t}$에 의해 특징지어진 선물 바스켓의 1달러에 대한 투자가치

4 Dealing With Multi-Product Series

- PCA 가중치
  - **목적**
    - $w_{i,t}$의 도출 방법 모색
    - V의 주성분 리스크 분포에 순응하는 배분 벡터 w 계산
  - **가정**
    - 크기 $N\ x\ I$, 평균 μ, iid 다변량 가우스 프로세스
    - 공분산 행렬 $N\ x\ N$
    - 주식 수익률, 채권 수익 변화, 옵션의 변동성 변화 등 랜덤 변수

4 Dealing With Multi-Product Series

- PCA 가중치
  - **방법**
    - $\boldsymbol{VW = W\Lambda}$
    - $\mathbf{V = W\Lambda W'}$
      - W열 Λ의 대각내림차순 정렬 재정의
    - **포트폴리오 리스크 계산**
      - given w
      - $\sigma^2 = w'Vw = w'W\Lambda W'w = \beta'\Lambda\beta = (\Lambda^{\frac{1}{2}}\beta)'(\Lambda^{\frac{1}{2}}\beta)$
    - $\boldsymbol{n}$**번째 성분 리스크**
      - $\sigma^2 = \sum\beta_n^2\Lambda_{\mathrm{n,n}}$
      - $R_n = \beta_n^2\Lambda_{\mathrm{n,n}}\sigma^2 = [W'w]_n^2\Lambda_{\mathrm{n,n}}\sigma^{-2}$
    - **사용자 정의 리스크 분포 R 산출 벡터 w 계산**
      - $\beta = \left\{\sigma\frac{\sqrt{R_n}}{\Lambda_{\mathrm{n,n}}}\right\}$이고, 이는 새로운 기저에서의 배분
    - **이전 기저에서의 배분**
      - $\boldsymbol{w = W\beta}$
  - **특징**
    - 대부분 주성분 분이 최대분산을 포함해 리스크 영향
    - PCA 포트폴리오는 최소-분산만 리스크 영향



4 Dealing With Multi-Product Series

- **단일 선물 롤**
  - **누적 롤 갭 시계열 형성 후 그 갭 계열만큼 가격에서 차감**
  - rollGaps 함수
    - 목적
      - 선행으로 롤 될 것인지 후행으로 롤 될 것인지 결정
        - 선행일 경우 원시 계열의 시작가
        - 후행일 경우 롤된 계열의 마지막 가격이 원시 계열의 마지막 가격과 매치
    - 인자
      - FUT_CUR_GEN_TICKER : 가격 연계 계약
      - PX_OPEN : 바 연계 시작가
      - PX_LAST : 바 연계 종가
      - VWAP : 바 연계 거래량-가중값 평균

```
def getRolledseries(pathIn ,key):
    #
    series=pd.read_hdf(pathIn ,key= "bars/ES_IOk")
    series["Time"]=pd.to_datetime(series["Time"], format= "%Y%m%d%H%M%Sf")
    series=series.set_index("Time")
    gaps=rollGaps(series)
    for f1d in ["close","VWAP"]: series[f1d]-=gaps
    return series

def rollGaps(series ,dictio={ "Instrument" : "FUT_CUR_GE_TICKER" ,
                              "Open" : "PX_OPEN" ,
                              "close" : "PlX_LAST"},
             matchEnd=True):
    # 이 전 종가와 다음 시가 사이에서 각 롤의 갭을 계산
    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] # 롤 이 전의 일의 인덱스
    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[-l] # 후방 롤
    return gaps
```

```
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 fid in [ "Open" , "close"]:rolled[fid]-=gaps
rolled ["Returns"] = rolled["close"] .diff()/ raw["close"].shift(l)
rolled ["rprices"] = (l+rol1ed ["Returns"]).cumprod()
```

4 Dealing With Multi-Product Series

- **단일 선물 롤**
  - **음이 아닌 롤된 계열**
    - **목적**
      - 투자 1 달러당 가격 계열 도출
    - **방법**
      - 1. 롤된 선물 가격에 시계열 계산
      - 2. 수익률 : 이전 원시 가격으로 나눈 롤된 가격 변화
      - 3. 이 수익률 을 사용해 가격 계열 구성

```
def getRolledseries(pathIn ,key):
    #
    series=pd.read_hdf(pathIn ,key= "bars/ES_IOk")
    series["Time"]=pd.to_datetime(series["Time"], format= "%Y%m%d%H%M%Sf")
    series=series.set_index("Time")
    gaps=rollGaps(series)
    for f1d in ["close","VWAP"]: series[f1d]-=gaps
    return series

def rollGaps(series ,dictio={ "Instrument" : "FUT_CUR_GE_TICKER" ,
                              "Open" : "PX_OPEN" ,
                              "close" : "PlX_LAST"},
             matchEnd=True):
    # 이 전 증가와 다음 시가 사이에서 각 롤의 값을 계산
    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] # 롤 이 전의 일의 인덱스
    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[-l] # 후방 롤
    return gaps
```

```
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 fid in [ "Open" , "close"]:rolled[fid]-=gaps
rolled ["Returns"] = rolled["close"] .diff()/ raw["close"].shift(l)
rolled ["rprices"] = (l+rol1ed ["Returns"]).cumprod()
```

05.

Sampling Features

5 Sampling Features

- 이전까지 구현한 데이터의 **머신러닝 적용시 한계**
  - 1. 머신러닝은 알고리즘은 표본 크기를 효율적으로 확장하지 못함
  - 2. 머신러닝 알고리즘은 유관된 데이터로부터 학습할 때 가장 높은 정확성
    - 특정 이벤트 후 분류기를 사용해 수익률 부호 예측시 더 정교한 예측 가능
- 이를 위해 **표본 추출해 연관된 훈련 예제**가 있는 **특징 행렬** 생성 필요

4 Dealing With Multi-Product Series

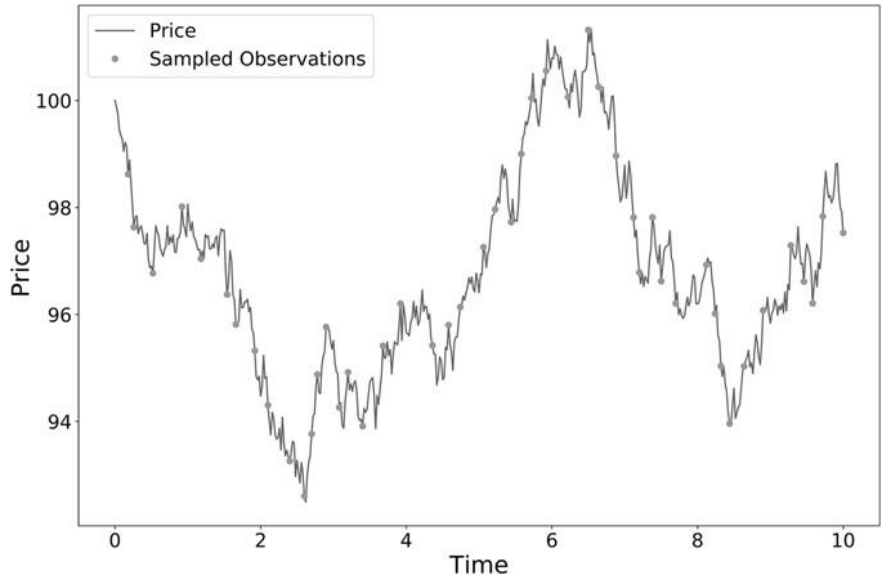
- **축소를 위한 표본 추출**
  - **목적**
    - 머신 러닝 알고리즘 적합화에 사용될 데이터양 줄임
      - 구조화된 데이터 세트로부터 특징 샘플링
    - **예시**
      - 순차적으로 표본 추출
        - **장점**
          - 단순함
        - **단점**
          - Seed에 따라 결과 편차가 큼
      - 유니폼 분포 랜덤 추출
        - **장점**
          - 보다 균일하게 표본 추출
        - **단점**
          - 여전히 표본이 정보 내용이나 예측력 관점에서 연관된 관측값을 포함하는게 보장되지 못함

4 Dealing With Multi-Product Series

- **이벤트 기반 표본 추출**
  - **목적**
    - 일반적으로 포트폴리오 관리자는 특정 사건 이후 베팅
    - 이런 이벤트들은 거시 경제 통계량과 연계되었을 경우 다분
    - 이벤트 구성 재정의, 혹은 다른 특징으로 예측 함수 생성 시도 필요

4 Dealing With Multi-Product Series

- **이벤트 기반 표본 추출**
  - CumSum 필터
    - **목적**
      - 품질 통제 기법
      - 측적값이 목표값의 평균으로부터 얼마나 벗어났는지 탐색
    - **가정**
      - 지역적으로 정적인 프로세스에서 발생한 IID 관측값
    - **정의**
      - **누적 합계**
        - $S_t = \max(0, S_{t-1} + y_t - E_{t-1}[y_t])$
          - 경계 조건 $S_0 = 0$
          - $y_t \leq E_{t-1}[y_t] - S_{t-1} \rightarrow S_t = 0$
        - **활성화**
          - $S_t \geq h \leftrightarrow \exists \tau \in [1, t] | \sum(y_i - E_{i-1}[y_t]) \geq h$
        - **대칭 CUMSUM 필터**
          - $S_t^+ = \max(0, S_{t-1}^+ + y_t - E_{t-1}[y_t])$, $S_0^+ = 0$
          - $S_t^- = \max(0, S_{t-1}^- + y_t - E_{t-1}[y_t])$, $S_0^- = 0$
          - $S_t = \max\{S_t^+, S_t^-\}$




Thank you
YONSEI UNIVERSITY