

PBL기반 금융빅데이터 분석가 과정



Finance 이론적 지식이나, 소개된 내용과 확장된 다양한 Finance 이론 소개 및 data 관련 Site 내용 이해를 위한 내용 등을 담습니다.

상당 부분은 E-book 내용이니 여러분들 학습에만 참고하시고 공유하시면 안되는 점 양지 바랍니다.

YB



Machine Learning for Algorithmic

Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python

using **Tensorflow**

텐서플로우를 이용한 주가 예측에서 가격-기반 입력 피쳐의 예측 성능 평가
(Performance Evaluation of Price-based Input Features in Stock Price Prediction
using Tensorflow)




Federal Reserve Economic Data

https://en.wikipedia.org/wiki/Federal_Reserve_Economic_Data



<https://pandas-datareader.readthedocs.io/en/latest/readers/fred.html>

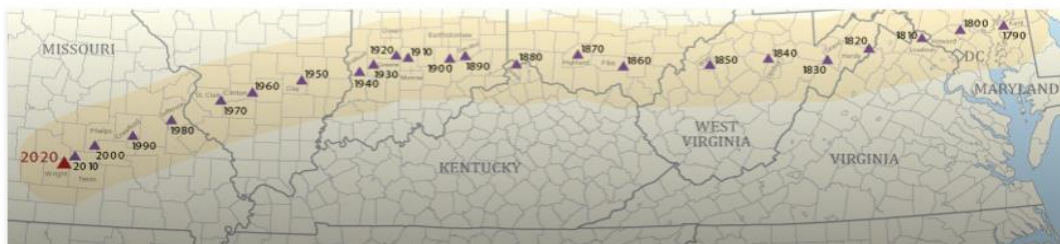
<https://www.census.gov/en.html>



Search

[BROWSE BY TOPIC](#)[EXPLORE DATA](#)[LIBRARY](#)[SURVEYS/ PROGRAMS](#)[INFORMATION FOR...](#)[FIND A CODE](#)[ABOUT US](#)

Small Business Pulse Survey Phase 7 Results are Now Available



<https://www.census.gov/quickfacts/fact/table/US/PST045219>

QuickFacts United States

QuickFacts provides statistics for all states and counties, and for cities and towns with a *population of 5,000 or more*.









Table

All Topics	United States
Population estimates, July 1, 2019, (V2019)	328,239,523
PEOPLE	
Population	
Population estimates, July 1, 2019, (V2019)	328,239,523
Population estimates base, April 1, 2010, (V2019)	308,758,105
Population, percent change - April 1, 2010 (estimates base) to July 1, 2019, (V2019)	6.3%
Population, Census, April 1, 2020	331,449,281
Population, Census, April 1, 2010	308,745,538
Age and Sex	

<https://www.census.gov/popclock/>

The U.S. population total and population change have been adjusted to be consistent with the results of the 2020 Census. The components of population change have not been adjusted and so inconsistencies will exist between population values derived directly from the components and the population displayed in the odometer and the Select a Date tool.

U.S. and World Population Clock



The United States



The World

Nov 30, 2021 04:08 UTC (+-9)

[Learn More](#) | [Download and Share](#)



U.S. Population

332,964,804



World Population

7,806,123,929

Components of Population Change

04:08:53 UTC

One birth every 8 seconds



One death every 11 seconds



One international migrant (net) every 649 seconds



Net gain of one person every 35 seconds



TOP 10 MOST POPULOUS COUNTRIES (July 1, 2021)

1. China	1,397,897,720	6. Nigeria	219,463,862
2. India	1,339,330,514	7. Brazil	213,445,417
3. United States	332,475,723	8. Bangladesh	164,098,818
4. Indonesia	275,122,131	9. Russia	142,320,790
5. Pakistan	238,181,034	10. Mexico	130,207,371

candlestick_ohlc 사용하기 (ch 4. page 165-166)

```
import datetime
import matplotlib.dates as mdates
import matplotlib.pyplot as plt
import pandas_datareader as pd
from mpl_finance import candlestick_ohlc
start = datetime.datetime(2018, 12, 1)
end = datetime.datetime(2018, 12, 31)
samsung = pd.DataReader("005930.KS", "yahoo", start, end)
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111)
# "from mpl_finance import candlestick_ohlc"
candlestick_ohlc(ax, zip(mdates.date2num(samsung.index.to_pydatetime()),samsung['Open'],samsung['High'],samsung['Low'],samsung['Close']),width=0.5,colorup="r",colordown="b")
plt.show()
```

Volatility

일간변동성(interday volatility)

일정기간 동안의 일별 주가(종가) 등락률의 표준편차를 의미

개념적으로만 본다면, 변동성과 가격상승 또는 하락은 관계가 없다.

일반적으로 시장에 위기가 오거나 가격이 하락할 것 같은 시기에 '변동성이 커진다'라는 표현을 사용

→ 일간 또는 일중 구분 없이 변동성을 말 할 때 '일간' 변동성을 의미 함. iid 가정

$$\text{일간변동성} = \sqrt{\frac{\sum_{t=1}^T (r_t - \bar{r})^2}{T}} \times \sqrt{250}$$

r_t : t 일의 수익률, \bar{r} : 1~T기간 중 (산술)평균수익률

일중변동성(intraday volatility)

하루 중에 주가가 얼마만큼 변동하는 할 수 있는지를 의미

즉, 장중 고가와 저가의 폭이 어느 정도 큰 지를 의미

종목에 호재 또는 악재가 발생하거나 투기적인 성향이 강한 종목일 수록 일중변동성이 높다.

$$t\text{일의 일중변동성} = \frac{(t\text{일중 고가} - t\text{일중 저가})}{\frac{(t\text{일중 고가} + t\text{일중 저가})}{2}}$$

$$\text{기간일중변동성} = \frac{\sum_{t=1}^T t\text{일의 일중변동성}}{T}$$

Volatility

사후적 변동성 (ex-post volatility)

투자성과가 발생한 후 결과를 확인하는 측면에서 측정하는 변동성

→ 역사적 변동성(Historical volatility)으로 이해

보통 변동성이라고 말하면 사후적 변동성을 의미 → 주식 수익률의 샘플 표준편차(Sample standard deviation)로 계산

$$\bar{r}_p = \frac{1}{m} \sum_{j=1}^m r_{pj}$$
$$\sigma_p = s_p = \left(\frac{1}{m} \sum_{j=1}^m (r_{pj} - \bar{r}_p)^2 \right)^{1/2}$$

사전적 변동성 (ex-ante volatility)

투자성과가 발생하기 전에 사전적으로 예측된 변동성

→ 공분산 행렬(Covariance matrix)로 추정

포트폴리오에서 현재 보유하고 있는 각 종목별 비중 벡터 $\mathbf{w}_p \in \mathbb{R}^d$ 및 종목별 수익률 확률변수 $\mathbf{X} \in \mathbb{R}^d$ 에 대해서, 포트폴리오 수익률의 분산(Variance)은 다음과 같이 계산

$$\mathbf{Var}[R_p] = \mathbf{Var}[\mathbf{w}_p^T \mathbf{X}] = \mathbf{w}_p^T \mathbf{\Sigma} \mathbf{w}_p$$

$$\mathbf{\Sigma} = \mathbf{Var}[\mathbf{X}] = \mathbf{Cov}[\mathbf{X}, \mathbf{X}] = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \cdots & \sigma_{dd} \end{bmatrix}$$

$\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$: 수익률의 공분산 행렬

사전적 변동성 σ_f 예측 =
(f = future 의미)

$$\sigma_f = \sqrt{\tau \mathbf{Var}[R_p]} = \sqrt{\tau \mathbf{w}_p^T \mathbf{\Sigma} \mathbf{w}_p}$$

Volatility 연구

- Harris(1986)과 Jain-Joh(1988)

뉴욕 주식시장의 수익률의 일중 변동성이 U자형 패턴을 보인다는 것을 발견

- WangWang(2010)

대만 주식시장의 수익률의 일중 변동성이 L자형 패턴을 보인다는 것을 발견

- 한국 주식시장의 경우 장하성(1992)

수익률의 일중 변동성 패턴을 분석한 결과 개장 시간대보다 폐장 시간대에 변동성이 더 큰 비대칭적 V자 형태가 존재함을 찾아내었다.

- 본 연구 과정의 대상이 되는 데이터를 이용하여 조사한 결과 KOSPI 시장 수익률의 일중 변동성 패턴은 개장 시간대가 폐장 시간대보다 오히려 큰 L자형에 가까운 U자형의 모습을 보인다는 것을 발견

Return

- **연간수익률 (APR : Annual Percentage Return)**

연 기준 수익률(연환산수익률), 흔히 시중에서 얘기되는 금리→ 복리를 고려하지 않는다.

ex) 분기 수익률이 2%인 경우 , $APR = 2\% * 4 = 8\%$ (연)

- **연간유효수익률(EAR : Effective Annual Return)**

→ 연환산수익률인데 복리 효과를 고려

ex) 분기수익률이 2%인 경우 , $EAR = (1.02)^4 - 1 = 8.243\%$ (연)

- **보유기간수익률(HPR : Holding Period Return)**

→ 투자 원금 대비 증권보유기간 동안 획득한 수익률, 대개 1년 기준

→ 주식의 경우 : 자본이득률(매매차익) + 배당수익률(D/P)

- **산술평균수익률 (arithmetic return)**

→ 기간 수익률(HPR)의 합을 기간 수로 나눈 것 (단리)

- **기하평균수익률(geometric return)**

→ 각 기간의 복리를 고려하여 계산한 수익률 , 항상 산술평균수익률보다 작거나 같다.

$(1+수익률)(1+수익률)$ 에 대한 제곱근 - 1

Adjusted Beta (ch5)

Table 1. Adjusted Beta Calculation Case

Source	Calculation Method
Bloomberg1	Adjusted Beta = $0.67 * \text{Raw Beta} + 0.33 * 1$
Merill Lynch2	Adjusted Beta = $0.67 * \text{Raw Beta} + 0.33 * 1$
Value Line3	Adjusted Beta = $0.67 * \text{Raw Beta} + 0.35 * 1$
Deloitte3	Adjusted Beta = $0.635 * \text{Raw Beta} + 0.371 * 1$

Source 1: Bloomberg The adjustment is estimated based on 67% confidence, or one standard deviation.

Source 2: Cost of Capital Technical Workshop, Draft Position Paper, 1988

Source 3: Ibbotson SBBI 2009 Valuation Yearbook

Asia-Pacific Journal of Business (아태 비즈니스 연구) Vol. 10, No. 4, December 2019 (pp.65-75)

시장효율적일수록 → 베타 1 수렴
역사적 베타 1보다 큰 경우,
1보다 작은 경우
네이버증권 : 52주 베타 (크롤링)

JOEL HASBROUCK*

THE JOURNAL OF FINANCE * VOL. L, NO. 4 * SEPTEMBER 1995 One Security, Many Markets:
 Determining the Contributions to Price Discovery JOEL HASBROUCK*

total long-run impact of the innovation

information share

$$S_{ij} = \frac{[\hat{\Psi}(1)C]_{ij}^2}{[\hat{\Psi}(1)\hat{\Omega}\hat{\Psi}(1)']_{ii}}$$

where, $\hat{\Omega} = CC'$

market micro structure

시장의 새로운 정보가 가격에 반영되어 가격이 실제로 실현되고 관찰되는 과정으로 정의
 시장의 가격발견 기능이란?

거래를 통해 형성되는 가격이 시장의 모든 정보와 자산의 실질가치를 완벽히 반영하기 위해 끊임없이 실현되는 과정
 → 자본주의 시장 경제에서 가격은 충분 통계량으로서 모든 의사결정 및 자원배분의 근거

→ 거래자간 가격정보의 부족으로 bid-ask spread가 증가→ 이에 따라 거래가 성립하지 않거나 거래가 성립하더라도 그 체결가는 시장 청산가격으로서의 적정가격이 아닐 가능성이 높아진다.

Normality test : shapiro-wilk test

Shapiro–Wilk test is a [test of normality](#) in frequentist [statistics](#). It was published in 1965 by [Samuel Sanford Shapiro](#) and [Martin Wilk](#).

CH 6. 유동성(liquidity)

유동성(liquidity)

시장미시구조나 자산가격결정, 기업 재무와 같은 다양한 재무분야에서 활용

예를 들면, 자산가격결정 분야에서는 유동성이 정적인 거래비용 정도의 개념이 아니라 시장전체적으로 공통적인 요인을 가지고 있고 이 공통요인에 대한 민감도에 따라서 위험요인(risk factor)으로까지 논의

Chordia, Roll and Subrahmanyam, 2000; Pastor and Stambaugh, 2003; Acharya and Pedersen, 2005; Sadka, 2006; Liu, 2006)

기업재무 분야에 서도 유동성이 기업의 자본구조나 발행주식의 형태, 자본조달의 결정 등에 영향을 줄 수 있음이 연구 (유동성 측정치 자료의 빈도수에 따른 구분)

- 고빈도 자료(high-frequency data) : intraday data

→ 일반 투자자들은 접근하기 어려우며, 존재하는 자료기간이 짧아서 사용할 수 있는 기간이 제한

- 저빈도 자료(low-frequency data): daily data

고빈도 자료를 사용한 스프레드 벤치마크

호가 스프레드율(Quoted Spread)

최우선 매도호가와 최우선 매수호가의 차이

스프레드는 즉시 거래하기 원하는 투자자가 지불해야 하는 비용

많은 시장미시구조의 연구들에서 **유동성의 측정치로서 스프레드 자료를** 이용

(Stoll and Whalley, 1983; Amihud and Mendelson, 1986, 1989; Eleswarapu and Reinganum, 1993; Kadlec and McConnell, 1994; Eleswarapu, 1997)

호가 스프레드는 최우선 매도호가에서 최우선 매수호가를 뺀 값

호가스프레드율은 호가스프레드를 최우선 매도호와 매수호가의 중간값으로 나눈 값

$$\text{호가스프레드율} = \frac{Ask_t - Bid_t}{(Ask_t + Bid_t)/2}$$

유효 스프레드율(Effective Spread)

유효스프레드는 거래가격과 거래 직전의 최우선 매도호와 매수호가의 중간값의 차이의 2배수이며, 유효스프레드율은 유효스프레드를 거래가격으로 나눈 값

$$\text{유효스프레드율} = 2 \times \frac{|P_t - (Ask_t + bid_t)/2|}{P_t}$$

저빈도 자료를 사용한 스프레드 측정치

A Simple Implicit Measure of the Effective Bid-Ask Spread in an Efficient Market

Richard Roll The Journal of Finance, Vol. 39, No. 4. (Sep., 1984), pp. 1127-1139

$$S = 2\sqrt{-cov(\Delta P_t, \Delta p_{t-1})}$$
$$\%spread = \frac{S}{\bar{P}}$$

Roll(1984) 스프레드 측정치는 주식시장에서의 체결 가격이 매 수/매도 호가가 반복됨으로 생길 수 있는 일별 주식수익률의 음의 자기상관현상을 이용하여 개별주식의 Roll 스프레드를 측정

S= Roll's spread P_t : 종가

Cov는 주식의 일별 수익률의 자기상관계수

저빈도 자료를 사용한 스프레드 측정치

Hasbrouck, J., “Trading costs and returns for US equities: Estimating effective costs from daily data”, *Journal of Finance*, Vol. 64(2009), pp. 1445-1477

Hasbrouck(2009)

Roll의 모형을 추정함에 있어서 Gibbs 샘플링 추정기법을 사용

$$\Delta P_t = c\Delta Q_t + e_t$$

P_t 는 주식의 t 시점의 거래가격을 의미

Q_t 는 주식의 t 시점의 거래방향을 나타내는 변수 → 매수거래이면 +1의 값을 매도거래이면 -1의 값

우리가 관심을 가지는 변수인 c 는 유효스프레드의 절반 값

Roll 모형에서 Q_t 가 +1이나 -1이 될 확률이 동일하고 계열상관이 없으며 e_t 와 독립적이라고 가정

Hasbrouck(2009)

위 식의 Roll의 모형을 추정함에 있어서 기존의 가정들을 사용하지 않고 Gibbs 샘플링 기법을 사용

그들은 단지 모형에서 e_t 가 0의 평균과 일정 분산을 지니는 정규분포를 따른다고 가정하고 나서, Gibbs 샘플링 기법을 통하여서 모수를 추정

** Gibbs sampling

두 개 이상의 확률 변수의 결합확률분포로부터 일련의 표본을 생성하는 확률적 알고리즘
베이저안 추론(Bayesian inference)에 근거하여 모수의 정상적 분포(stationary distribution)를 찾는다.

정상적 분포를 구하기 위하여 마르코프 체인(Markov Chain)을 따르는 추출값들(draws)을 이용하며 이 때 추출값을 구하는 방법이 몬테카를로시뮬레이션(Monte Carlo simulation)이다.

→ 깁스 샘플링의 가장 큰 장점은 고도의 복잡한 모형도 추정할 수 있다는 점

마코프 연쇄(Markov Chain)

: 마코프 가정(Markov assumption)을 따르는 이산 시간 확률과정으로 마코프 가정은 러시아 수학자 마코프가 1913년경에 러시아어 문헌에 나오는 글자들의 순서에 관한 모델을 구축하기 위해 제안된 개념으로 특정 시점의 상태 확률은 단지 그 이전 상태에만 의존한다는 것이 핵심

- 한 상태에서 다른 상태로의 전이(transition)는 그동안 상태 전이에 대한 긴 이력(history)을 필요로 하지 않고 바로 직전 상태에서의 전이로 추정할 수 있다는 것.
- 특정 조건을 만족한 상태에서 마코프 연쇄를 반복하다 보면 현재 상태의 확률이 직전 상태의 확률과 같아지게, 수렴하게 되므로 이렇게 평형 상태에 도달한 확률 분포를 정적분포(Stationary Distribution)라고 한다.

마코프 연쇄 몬테카를로 방법(MCMC)

MCMC란 마코프 연쇄에 기반한 확률 분포로부터 표본을 추출하는 몬테카를로 방법으로 MCMC 알고리즘은 우리가 샘플을 얻으려고 하는 목표분포를 Stationary Distribution으로 가지는 마코프 체인을 만든다. 이 체인의 시뮬레이션을 가동하고 초기값에 영향을 받는 burn-in period를 지나고 나면 목표분포를 따르는 샘플이 만들어진다.

MCMC 알고리즘은 우리가 샘플을 얻으려고 하는 목표분포를 Stationary Distribution으로 가지는 마코프 체인을 만든다.

이 체인의 시뮬레이션을 가동하고 초기값에 영향을 받는 burn-in period를 지나고 나면 목표분포를 따르는 샘플이 만들어진다.

깁스 샘플링(Gibbs sampling)

몬테카를로 방법은 모든 샘플이 독립(independent)이고 생성될 확률 또한 랜덤이지만. 반면 마코프 연쇄에 기반한 MCMC는 다음번 생성될 샘플은 현재 샘플의 영향을 받지만,

깁스 샘플링은 다음번 생성될 표본은 현재 샘플에 영향을 받는다는 점에서는 MCMC와 같지만, 나머지 변수는 그대로 두고 한 변수에만 변화를 준다는 점이 차이가 있다.

깁스 샘플링 함수

깁스 샘플링의 변형들

변수가 (a,b,c) 세 개인 데이터에 대해 깁스 샘플링을 수행한다면 b,c 를 고정시킨 채로 a 를, a,c 를 고정시킨 채로 b 를, a,b 를 고정시킨 채로 c 를 차례대로 뽑아야 한다.

Block Gibbs sampling 기법은 그룹으로 묶어 뽑는 기법

c 를 고정시킨 채로 a,b 를 뽑고 a,b 를 고정시킨 채로 c 를 뽑는 방식

Collapsed Gibbs sampling 기법은 불필요한 일부 변수를 샘플링에서 생략하는 기법

b 가 그런 변수라 가정하면 c 를 고정시킨 상태에서 a 를 뽑고, a 를 고정시킨 상태에서 c 를 뽑는다.

Python 사례:깁스 샘플링 함수

- 첫번째 주사위의 눈을 x ,
- 두 주사위의 눈을 합한 값을 y
- x 와 y 의 결합확률분포 함수

```
import random
def roll_a_die():
    # 주사위 눈은 1~6
    # 각 눈이 선택될 확률은 동일(uniform)
    return random.choice(range(1,7))
def direct_sample():
    d1 = roll_a_die()
    d2 = roll_a_die()
    return d1, d1+d2
def random_y_given_x(x):
    # x값을 알고 있다는 전제 하에
    # y값이 선택될 확률
    # y는 x+1, x+2, x+3
    # x+4, x+5, x+6 가운데 하나
    return x + roll_a_die()
```

x 에 대한 y 의 조건부확률과 y 에 대한 x 의 조건부확률 함수

```
def random_x_given_y(y):
    # y값을 알고 있다는 전제 하에
    # x값이 선택될 확률
    # 첫째 둘째 주사위 값의 합이 7이거나
    # 7보다 작다면
    if y <= 7:
        # 첫번째 주사위의 눈은 1~6
        # 각 눈이 선택될 확률은 동일
        return random.randrange(1, y)
    # 만약 총합이 7보다 크다면
    else:
        # 첫번째 주사위의 눈은
        # y-6, y-5,..., 6
        # 각 눈이 선택될 확률은 동일
        return random.randrange(y-6, 7)
```

깁스 샘플링 함수

```
def gibbs_sample(num_iters=100):
    # 초기값이 무엇이든 상관없음
    x, y = 1, 2
    for _ in range(num_iters):
        x = random_x_given_y(y)
        y = random_y_given_x(x)
    return x, y
```

회전율(Turnover)

회전율 = 거래량 / 상장주식수

투자자들이 얼마나 많은 양을 거래했는 지를 측정

투자자들이 평균적으로 전체 주식수에 대해서 얼마나 빨리 그들의 포지션을 전환시키는 지를 나타내 준다.

회전율은 일반적인 투자자의 보유기간(holding period)의 역수의 개념으로 사용되기도 한다.

Chordia, Subrahmanyam, and Anshuman(2001)은 거래금액과 회전율을 유동성의 측정치로 사용하여서 자산가격결정 검증을 실시하였으며, **회전율이 주식 수익률과 유의적인 음(-)의 관계를 지니고 있음을 발견**

한국시장에서 윤상용, 구본일, 엄영호, 한재훈(2009)은 거래회전율을 이용하여 유동성위험 모방포트폴리오를 구성하여 주식수익률을 검증함으로써 유동성 요인이 한국주식시장에서 설명력이 있음을 보여주었다.

고빈도 자료를 사용한 가격충격(Price Impact) 벤치마크

Hasbrouck, J., “Trading costs and returns for US equities: Estimating effective costs from daily data”, Journal of Finance, Vol. 64(2009), pp. 1445-1477

Habrouck (2009)의 Lambda(λ)

Hasbrouck (2009)는 5분의 거래기간의 수익률을 동일 기간 동안의 거래방향이 표시된 거래 금액의 제곱근(signed square-root dollar volume)에 대해서 회귀분석을 실행함으로써 가격충격의 측정치를 구하였다.
다음과 회귀식을 추정함을 통하여 회귀계수 Lambda(λ)를 계산

$$r_n = \lambda \left[\sum_t \text{sign}(\text{volume}) \sqrt{|\text{volume}_{t,n}|} \right] + u_n$$

- r_n : n번째 5분 기간 동안의 수익률
- $\text{volume}_{t,n}$: n번째 5분 기간 동안에 해당되는 거래들 중에서 t번째 거래의 거래금액
- $\text{sign}(\cdot)$: t번째 거래가 매수거래 +1 , 매도거래 -1.
- U_n : 교란항

저빈도 자료를 사용한 가격충격(Price Impact) 측정치

Amihud, Y., “Illiquidity and stock returns: cross-section and time-series effects”, Journal of Financial Markets, Vol. 5(2002), pp. 31-56.

Amihud(2002) 비유동성의 측정치 제안

$$Amihud_t^i = \frac{1}{Days_t^i} \sum_{d=1}^{Days_t^i} \frac{|R_{td}^i|}{V_{id}^i}$$

어떤 주식이 작은 거래량에 비해서 가격 변화가 크다면 이 주식은 큰 Amihud 측정치의 값을 갖게 될 것이며, 이는 그 주식이 비유동적이라는 것을 의미한다.

Amihud(2002)는 시간에 따라서 기대되는 시장의 비유동성이 미래의 초과수익률과 유의적인 양(+)의 관계를 가지고 있음을 발견하였으며, 이 현상을 근거로 주식의 기대수익률이 부분적으로는 유동성 프리미엄을 반영하고 있음을 주장

CAPM검증 모형

시계열분석에 의한 사전적(ex ante)인 모형 CAPM검증 모형

$$E(R_{jt}) = R_{ft} + \beta_j[E(R_{mt}) - R_{ft}]$$

사후적(ex post)모형으로 변형

$$(R_{Pt} - R_{ft}) = \alpha_P + \beta_P(R_{mt} - R_{ft}) + e_{Pt}$$

전통적 CAPM이 현실적으로 성립하는지 검증하기 위해서는 다음의 식 같은 귀무가설과 대립가설을 검증

$$H_0 : \alpha_P = 0$$

$$H_1 : \alpha_P \neq 0$$

➔ 통계적 의미로 0이 될 때 전통적 CAPM이 현실적으로 성립한다고 볼 수 있다

베타 추정

개별증권들의 베타 계수를 시장모형에 의하여 추정

$$R_{jt} = \alpha_j + \beta_j R_{mt} + e_{jt} \quad \text{Beta coefficient}(\beta) = \frac{\text{Covariance}(R_e, R_m)}{\text{Variance}(R_m)}$$

- 베타계산은 과거 일정기간의 월별 주식수익률과 월별시장수익률의 자료를 이용
- 네이버 증권 ; 52주 베타 제공(역사적 베타)
- 조정 베타

$$\beta_{port} = \sum_{i=1}^n w_i \beta_i$$

Fama·MacBeth(1973)의 횡단면 회귀분석 검증 접근법

Fama, Eugene F. and James MacBeth, 1973, Risk, return and equilibrium: Empirical tests, *Journal of Political Economy* 81, 607-636

Fama & MacBeth(1973)의 방법론은 3단계 접근법을 수정한 검증 방법

제1단계

표본기간을 하위표본기간 15년씩으로 분류한다. 이 기간을 각각 5년씩 포트폴리오 구성기간, 포트폴리오 베타 추정기간, 그리고 검증기간으로 나눈다. 개별증권 베타의 계산은 포트폴리오 구성기간에 **시장모형을 이용하여** 각각의 증권에 대하여 베타들을 추정 → 추정된 베타의 크기에 따라 증권들을 정렬한 다음, 동일한 기업수로 10개의 포트폴리오들을 구성 → 가장 높은 베타들로 구성된 증권들을 첫번째 포트폴리오로 하고, 가장 낮은 베타들로 구성된 증권들을 마지막 10번째 포트폴리오로 한다.

제2단계

두 번째 5년의 기간 동안에 포트폴리오의 수익률 포트폴리오를 구성하고 있는 증권들의 동일 가중수익률을 종속변수로 하고 시장수익률을 설명변수로 하여 아래 회귀분석을 이용하여 포트폴리오들의 베타들을 추정한다.

$$R_{pt} = \alpha_p + \beta_p R_{mt} + \varepsilon_{pt},$$

제3단계

포트폴리오의 베타와 주식수익률 간의 관계를 검증하는 단계로, 제2단계에서 추정된 15개의 포트폴리오의 베타와 포트폴리오에 속하는 개별증권의 1개월간의 주식수익률을 **횡단면 회귀분석을 실시하여** 아래 식의 회귀계수 γ_1 을 각각 추정한다.

→ 한국 주식시장에서 양(+)의 위험-수익률간의 상충관계 (trade-off)를 검증할 수 있다.

$$R_{it} = \hat{\gamma}_{0t} + \hat{\gamma}_{1t} \cdot \beta_i + \varepsilon_{it},$$

Fama–MacBeth regression

1. First regress each of n asset returns against m proposed risk factors to determine each asset's beta exposures.

$$\begin{aligned}R_{1,t} &= \alpha_1 + \beta_{1,F_1} F_{1,t} + \beta_{1,F_2} F_{2,t} + \cdots + \beta_{1,F_m} F_{m,t} + \epsilon_{1,t} \\R_{2,t} &= \alpha_2 + \beta_{2,F_1} F_{1,t} + \beta_{2,F_2} F_{2,t} + \cdots + \beta_{2,F_m} F_{m,t} + \epsilon_{2,t} \\&\vdots \\R_{n,t} &= \alpha_n + \beta_{n,F_1} F_{1,t} + \beta_{n,F_2} F_{2,t} + \cdots + \beta_{n,F_m} F_{m,t} + \epsilon_{n,t}\end{aligned}\quad [1]$$

2. Then regress all asset returns for each of T time periods against the previously estimated betas to determine the risk premium for each factor.

$$\begin{aligned}R_{i,1} &= \gamma_{1,0} + \gamma_{1,1} \hat{\beta}_{i,F_1} + \gamma_{1,2} \hat{\beta}_{i,F_2} + \cdots + \gamma_{1,m} \hat{\beta}_{i,F_m} + \epsilon_{i,1} \\R_{i,2} &= \gamma_{2,0} + \gamma_{2,1} \hat{\beta}_{i,F_1} + \gamma_{2,2} \hat{\beta}_{i,F_2} + \cdots + \gamma_{2,m} \hat{\beta}_{i,F_m} + \epsilon_{i,2} \\&\vdots \\R_{i,T} &= \gamma_{T,0} + \gamma_{T,1} \hat{\beta}_{i,F_1} + \gamma_{T,2} \hat{\beta}_{i,F_2} + \cdots + \gamma_{T,m} \hat{\beta}_{i,F_m} + \epsilon_{i,T}\end{aligned}\quad [1]$$

demonstrated that the residuals of risk-return regressions and the observed "fair game" properties of the coefficients are consistent with an "efficient capital market" (quotes in the original).^[2]

Note that Fama MacBeth regressions provide [standard errors](#) corrected only **for cross-sectional correlation**. The standard errors from this method do not correct for time-series autocorrelation. This is usually not a problem for stock trading since stocks have weak time-series autocorrelation in daily and weekly holding periods, but autocorrelation is stronger over long horizons.^[3] This means Fama MacBeth regressions may be inappropriate to use in many corporate finance settings where project holding periods tend to be long. For alternative methods of correcting standard errors for time series and cross-sectional correlation in the error term look into double clustering by firm and year

CH8. 회귀분석 ; 변수 선택 방법(variable selection)

회귀모델에서 설명변수가 많을 경우에는 설명변수를 줄일 필요가 있다.

→ 설명변수가 많으면 예측 성능이 좋지 않기 때문이다..

여러 설명변수를 가지는 회귀분석의 경우 설명변수들 사이의 독립성 등의 가정을 만족시키기 어렵고. 또한 설명변수의 증가는 모형의 결정계수 등을 증가시키기는 하지만 다중 공선성 문제 등을 일으키므로 결과적으로 추정의 신뢰성을 저하 문제 발생

1. stepwise

- forward selection
- backward elimination
- stepwise selection

2. Shrinkage

Lasso Regression (L1 Regression)

Ridge Regression (L2 Regression)

3. Dimension Reduction

(판단 지표)

- adjusted 결정계수,
- **AIC(Akaike Information Criterion)**
- **BIC(Bayes Information Criterion)**

CH8. AIC(Akaike information criterion, SBC(Schwarz-Bayesian Information Criterion), BIC

일본 통계학자 [Hirotugu Akaike](#) 의 이름을 따서 명명

주어진 데이터 셋에 대한 통계 모델의 상대적인 품질을 평가하는 것

→ AIC는 주어진 모델에서 손실되는 정보의 상대적인 양을 추정하므로 모델이 손실하는 정보가 적을수록 해당 모델의 품질이 높아진다.

모델에 의해 손실된 정보의 양을 추정할 때 AIC는 모델의 [적합도](#)와 모델의 단순성간의 균형을 다룬다.

$$AIC = 2k - 2\ln(\tilde{L})$$

2k는 모형의 추정된 parameter 개수 → 해당 모형에 패널티를 주기 위해 사용

→ 실제로 어떤 모형이 $2\ln(L)$ 즉 적합도를 높이기 위해 여러 불필요한 파라미터를 사용할 수도 있다. 실제 모형 비교 시 독립변수가 많은 모형이 적합도 면에서 유리하게 되는데, 이는 독립변수에 따라서 모형의 적합도에 차이가 난다는 의미이므로 이를 상쇄시키기 위하여 불필요한 파라미터, 즉 독립변수 수가 증가할수록 2k를 증가시켜 패널티를 부여하여 모형의 품질을 평가하는 의미로 이해하면 됨.

-2ln(L)

모형의 적합도를 의미 : -2ln(L)에서 L은 Likelihood function 을 의미

→ AIC 값이 낮다는 것은 즉 모형의 적합도가 높은 것을 의미

SBC(Schwarz-Bayesian Information Criterion), BIC

$$SBC = k\ln(n) - 2\ln(L^*)$$

n : data 개수

(*)가 붙은 것은 이미 최적화가 된 어떤 상수(constant)라는 의미

$2\ln(L^*)$ 는 적합도(goodness of fit)를 대변해주는 수 → 이 수가 커지면 AIC와 SBC는 값이 감소

→ 2k나 $k\ln(n)$ 은 값이 커지면 AIC와 SBC가 커진다.

→ AIC나 SBC의 값이 크면 좋지 못한 모형이기 때문에 이 부분은 벌점요소(penalty)

→ 어떤 모형이 가장 적절한지 판단을 하기 위해 $2\ln(L^*)$ 를 이용해서 값을 추정하더라도 이 값을 높이기 위해 실제로 불필요하게 모형을 복잡하게 많은 parameter로 구성할 수도 있다. 이러한 경우를 방지하기 위해 이런 벌점요소(penalty) 주어 짐.

- **Skewness(비대칭도, 왜도)** : 어떤 확률변수의 분포가 평균을 중심으로 얼마나 비대칭인가를 나타내는 척도 (**정규분포 Skewness = 0**)
- **Kurtosis(첨도)** : 어떤 확률변수의 분포가 얼마나 뾰족한가를 나타내는 척도 (**정규분포 Kurtosis=3**)

Jacque-Bera (JB) Normality test

가설검정과 구간 추정에 있어서 중요한 전제 귀무가설과 대립가설

: $H_0: X \sim N(\bar{X}, \sigma^2)$, $H_1: H_0$ is not true

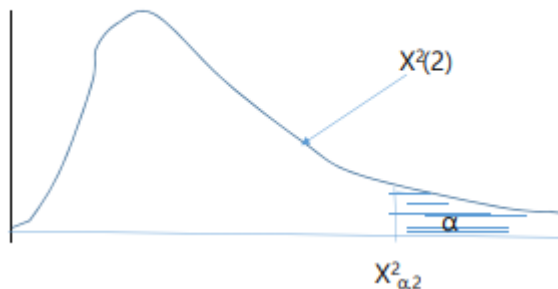
검정통계량

$$JB \equiv \frac{T}{6} \left(\hat{s}^2 + \frac{(\hat{k} - 3)^2}{4} \right) \sim \chi^2(2) : \text{under } H_0$$

JB

X가 정규분포라면 그 경험적 비대칭도와 경험적 첨도는 각각 0과 3에 가까운 값일 것이므로 JB의 값은 0에 가까운 값이 됨. 따라서 JB가 0보다 충분히 큰 경우, 즉 오른쪽 꼬리의 값이 나올 때 귀무가설을 기각한다.

기각역



기각할 수 없을 때, X의 분포가 정규분포라고 볼 수 있음

Ch9. Portfolio Theory

combination of two or more assets

Harry M. Markowitz : 포트폴리오이론의 기초 최초 제공
"Portfolio Selection", *Journal of Finance* (1952)

Investing is a trade-off between risk and expected return

For a given amount of risk, MPT describes how to select a portfolio with the highest possible expected return.

Or, for a given expected return, MPT explains how to select a portfolio with the lowest possible risk

$$\begin{aligned} E(r_p) &= w_1 \cdot E(r_1) + w_2 \cdot E(r_2) + \cdots + w_n \cdot E(r_n) \\ &= \sum_{i=1}^n w_i \cdot E(r_i) \end{aligned}$$

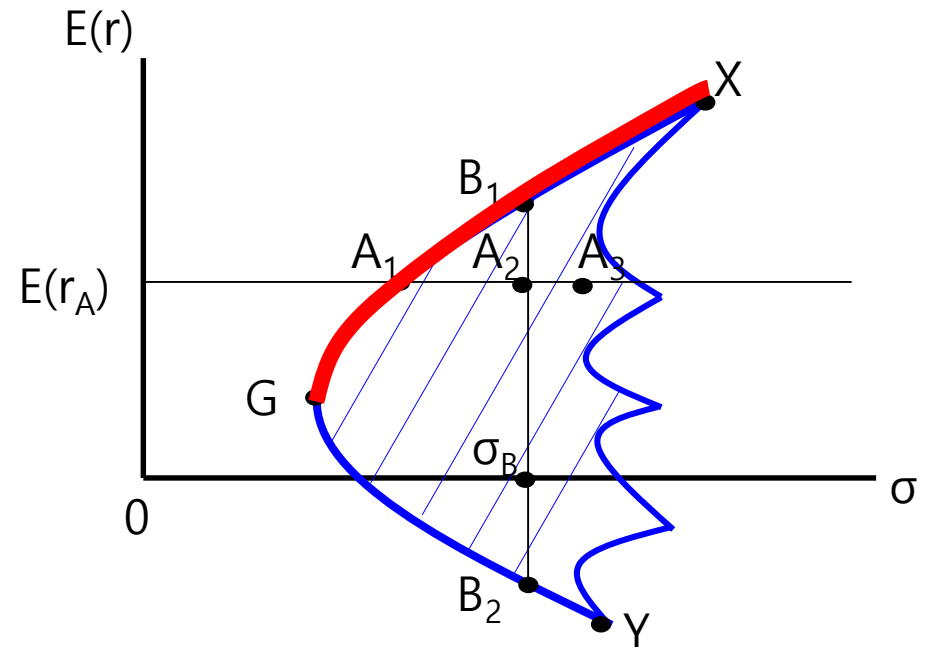
가중평균

$$\begin{aligned} \sigma_P^2 &= \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij} \\ &= \sum_{i=1}^n \sum_{j=1}^n w_i w_j \rho_{ij} \sigma_i \sigma_j = \sum_{i=1}^n w_i^2 \cdot \sigma_i^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij} \\ \sigma_P^2 &= w_1 \cdot \sigma_{1P} + w_2 \cdot \sigma_{2P} + \cdots + w_n \cdot \sigma_{nP} \\ &= \sum_{i=1}^n w_i \cdot \sigma_{iP} \end{aligned}$$

포트폴리오효과 (portfolio effect) : 자산을 결합하여 포트폴리오를 구성함으로써 위험이 줄어들어 기대효용이 증가하는 현상 → diversification effect

CH9 : Efficient frontier

- H. Markowitz 효율적 투자선 (1952)
- 투자기회집합 전체에서 지배원리를 충족시키는 포트폴리오의 집합
- 합리적 투자자는 효율적 투자선에 오는 포트폴리오를 선택
- 최적증권의 선택은 효율적투자선 상에 오는 증권중에서 무차별곡선과 접점(효용 극대화) 증권을 선택



Ch10. Option & futures

- American option : 권리행사 시기
- European option : KRX option 콜= 기초자산가격- 행사가격
- Implied Volatility(IV)

→ 옵션의 시장가격으로부터 Black-Scholes model 을 이용하여 변동성을 계산할 수 있는데 이를 내재변동성 (Implied Volatility : IV) 이라 말한다. 즉 옵션 현재 시장가격에 내재되어 있는 변동성을 말한다. → VIX (cboe, s\$500옵션)
→ (historical volatility, HV) : 과거의 특정기간 동안(통산 최근 1개월)의 기초자산수익률 변동성이 역사적 변동성 (historical volatility) 혹은 통계적 변동성 (statistical volatility, SV)이라고 한다. → 사후적 변동성

Black-Scholes-Merton model

$$\begin{aligned}c &= S_0 N(d_1) - X e^{-rT} N(d_2) \\p &= X e^{-rT} N(-d_2) - S_0 N(-d_1)\end{aligned}$$

Put-call parity

$$C + X e^{-r_f T} = P + S_0$$

$$\begin{aligned}d_1 &= \frac{\ln\left(\frac{S_0}{X}\right) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} \\d_2 &= d_1 - \sigma\sqrt{T}\end{aligned}$$

Ch11. Value at Risk . jorion

For a given portfolio, time horizon, and probability

VaR can be defined informally as the **maximum possible loss**

It estimates how much a set of investments might lose (**with a given probability 95%, 99%**), given normal market conditions, in **a set time period such as a day**.

VaR is typically used by firms and regulators in the financial industry to gauge the amount of assets needed to cover possible losses.

VaR 측정방법

- **Parametric method** – Also referred to as the **variance-covariance method**, the parametric method assumes a normal distribution of returns. This means that you only need to estimate two factors, the expected return and the standard deviation, which allows you to plot a normal distribution curve. This value at risk formula is best suited to cases where you can reliably estimate the distributions.
- **Historical method** – With the historical method, you'll essentially re-organise real historical returns by ranking them from worst to best. For example, if you use market data from the past 100 days, the second-worst day will be 99% VaR, the third-worst day will be 98% VaR, and so on.
- **Monte Carlo method** - With the Monte Carlo method, you can calculate value at risk using non-linear pricing models to randomly create different scenarios for future rates. Then, you can calculate VaR by estimating the change in value for each scenario and looking at the worst losses.

variance-covariance method

$$VaR = p * z * \sigma$$

Ch11. Stress Testing

금융에서의 금융시스템 스트레스 테스트(financial system stress test)

금융기관의 재무건전성을 평가하는 방법

예외적이지만 일어날 수 있는 가능성이 있는 여러 시나리오를 상정하여 그러한 위기 상황시 해당 금융기관의 재무 건전성을 파악하는 것

-1990년대 초부터 국제투자은행들이 고안하여 사용하기 시작한 리스크 관리기법으로 극단적으로 악화된 경제상황에서 투자은행들이 직면하게 되는 영업중단위험을 계량화하는 방법이다. 개별금융기관에 적용하는 스트레스 테스트의 개념을 총량적 차원에서 금융시스템으로 확장시킨 것

(목적)

특정한 시나리오 하에서 전체 금융기관의 잠재적 손실규모를 측정하는 정책당국 및 감독당국은 금융시스템 스트레스 테스트를 통해 금융시스템의 전반적인 리스크 익스포저 상황 및 그 추세를 파악하여 이를 금융시스템의 안정성을 평가하는데 활용

(적용 예시).

- 실업률이 X% 증가하면 어떤 변화가 얼마나 일어날 있을 것인가?
- GDP가 X% 하락하면 어떤 변화가 얼마나 일어날 있을 것인가?
- 이자율이 X% 상승하면 어떤 변화가 얼마나 일어날 있을 것인가?
- 원유가격이 X% 상,하락하면 어떤 변화가 얼마나 일어날 있을 것인가?

이런 가상의 위기 상황에 따른 재무제표 모의 결과는 개별 금융기관만이 당하는 어려움만 보여줄 뿐, 전체 금융 시장(금융기관 서로가 서로에게 채권자이자 채무자인 복잡한 시스템)이 혼란에 처하였을 경우에 대해서는 제대로 된 답을 줄 수는 없다.

金融시스템의 스트레스 테스트 方案 - 신용위험을 중심으로 -金 周 哲* (한국은행 금융경제연구원 : 참고

Ch 12. Monte Carlo Simulation

반복된 무작위 **추출**(repeated random sampling)을 이용하여 함수의 값을 수리적으로 근사하는 **알고리즘**을 부르는 용어

Monte Carlo Simulation 과정

문제정의 → 변수 선택 → 자료 수집 → 확률분포 선택 → random number 발생 → simulation → Modeling

1. 문제정의 : 불확실한 상황 하에서 의사결정, 문제 정의
2. 확률변수 선택(X_1, X_2, \dots, X_n)
3. 자료 수집
4. 확률변수의 확률 분포 선택
→ $P(X)$ 가 가질 수 있는 분포에 대한 선정을 통해 정확도 결정 : 정규분포, 로그정규분포 등
5. 확률변수의 random number 생성
6. Simulation 실시
: 생성한 난수를 식에 대입하여 시뮬레이션 실험을 실시, → 반복을 통해 정확한 예측치 획득
7. 통계량 계산 → 결과 해석

미국의 원자폭탄 개발 계획 : 맨해튼 프로젝트

- 몬테카를로 방법이라는 이름이 처음 쓰이게 된 계기
- 폴란드 출신의 수학자 **스타니스로 울람**(Stanisław Marcin Ulam, 1909-1984)
- 컴퓨터의 아버지로 잘 알려진 **폰 노이만**(Johann Ludwig von Neumann, 1903-1957) 등과 함께 맨해튼 프로젝트에 참여



몬테카를로 방법을 처음 명명한 수학자
스타니스로 울람 (c) 위키미디어

구글 딥마인드의 알파고는 이세돌 9단과의 바둑대결

컴퓨터가 스스로 학습하는 **딥러닝**(Deep Learning)과 함께, 몬테카를로 방법을 적용한 **알고리즘**

알파고는 '정책망'과 '가치망'이라 불리는 2개의 신경망으로 구성되었는데, 정책망이 다음 번 돌을 놓을 여러 경우의 수를 제시하면, 가치망은 그중 가장 적합한 한 가지 예측치를 제시하는 역할을 한다.

이 과정에서 모든 경우의 수를 다 계산하는 것은 불가능하므로, **표본을 추출하여 승률을 어림잡는 몬테카를로 방법을 적용하는 것으로 알려짐.**

Ch13. Credit Risk Analysis

<https://www.moodyys.com/>

<https://www.spglobal.com/ratings/en/>

<https://www.fitchratings.com/>

<https://content.naic.org/>

NAIC is the U.S. standard-setting and regulatory support organization created and governed by the chief insurance regulators.

Ch13. Credit Risk Analysis

<https://monevator.com/bond-default-rating-probability/>

Cumulative Historic Default Rates (in percent)

Rating category	Moody's		S&P	
	Muni	Corp	Muni	Corp
Aaa/AAA	0.00	0.52	0.00	0.60
Aa/AA	0.06	0.52	0.00	1.50
A/A	0.03	1.29	0.23	2.91
Baa/BBB	0.13	4.64	0.32	10.29
Ba/BB	2.65	19.12	1.74	29.93
B/B	11.86	43.34	8.48	53.72
Caa-C/CCC-C	16.58	69.18	44.81	69.19
Averages				
Investment grade	0.07	2.09	0.20	4.14
Non-investment grade	4.29	31.37	7.37	42.35
All	0.10	9.70	0.29	12.98

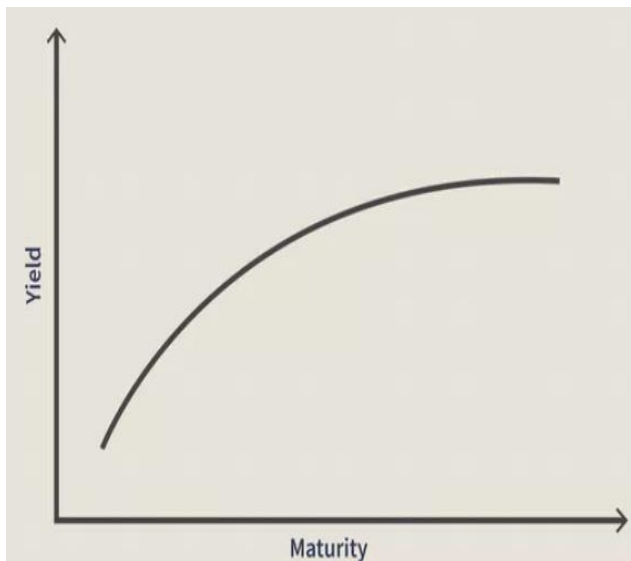
Source: U.S. Municipal Bond Fairness Act, 2008.

If you'd still like to add **non-investment grade bonds** to your portfolio to benefit from their higher yield, **I'd strongly suggest you avoid buying** them directly, given these **significant risks of default**.

Ch13. Credit Risk Analysis

Term Structure Of Interest Rates (yield)

The term structure of interest rates, commonly known as the yield curve, depicts the interest rates of similar quality bonds at different maturities.

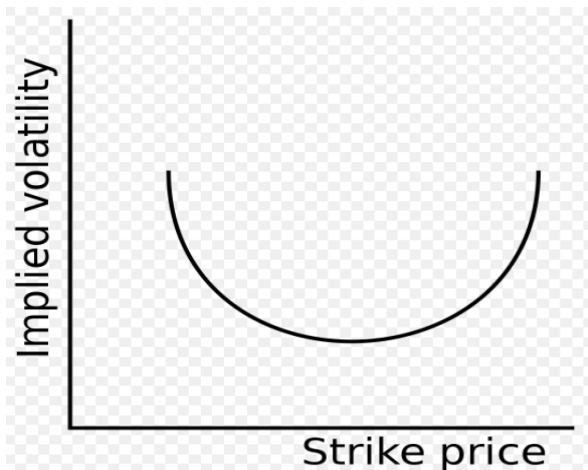


Ch15. Volatility, Implied Volatility, ARCH and GARCH

Volatility

Implied Volatility (IV)

- **Volatility Smile** : 내재변동성(IV)을 그래프로 나타내 보면 사람이 미소 짓고 있는 모양으로 ITM, OTM 옵션 IV 가 높고, ATM 옵션 IV 옵션이 낮은 현상
- **Volatility Smirk** : IV 가 ITM 옵션이 가장 낮게 나타나는 현상
- ITM , ATM($S=K$) , OTM



https://en.wikipedia.org/wiki/Volatility_smile

Ch15. Volatility, Implied Volatility, ARCH and GARCH

ARCH and GARCH

주가는 [랜덤워크](#)에 가깝게 움직이므로 내일의 코스피 지수를 예측하는 경우 '오늘 코스피 지수 +- 무작위 오차'식으로 예측 할 수 있다.

지수의 변동성(위험)이 군집성(volatility clustering) 특성을 갖고 있다.

→ 만약 시장이 박스권에 있으면 내일도 변동성이 작을 것이고,

→ 급락,급등장에서는 또 급락/급등이 이어질 확률이 높다는 것이다.

따라서 지수 자체를 예측하는것은 힘들지만 그 변동성을 예측하기 위해 나온 모형이 ARCH, GARCH 모형이다.

작은 변화는 당분간 지속적으로 작은 변화를 나타낸다는 것으로 이러한 경향

변동 집중성(volatility clustering)인데 이러한 변동 집중성에 의한 변화 폭은 제곱급이나 로그 변환으로 변동 집중성이 사라지지 않는다.

→ 적절한 시계열 모형(특히 AR 모형)을 적용하여야 설명될 수 있는데

→ 이와 같이 시계열 모형을 ARCH(autoregressive conditional heteroscedasticity) 모형이라고 한다.

→ 공적분(cointegration)에 대한 연구 공로로 노벨경제학상을 탄 C. Granger와 R. Engle 중 Engle이 만든 모형

시계열 데이터 자체보다는 해당 시계열의 변동성의 분석 및 예측을 위한 모형으로 시계열 자료의 오차항, 조건부분산 등을 이용하여 모형을 추정한다.

→ nonnegativity 조건 (모두 성분 모두가 음이 아닌 "Nonnegative 행렬") 이 성립하지 않을 위험성 등이 있기 때문에 이를 보완하기 위해 만들어진 것이 GARCH 모형이고, 실제로 ARCH(∞) 모형이 GARCH(1, 1)과 동치이기 때문에 적은 파라미터로 동일하거나 더 나은 설명력을 보여준다.

→ 이러한 이유로 거의 모든 논문들이 GARCH나 GARCH를 변형한 모형으로는 IGARCH, EGARCH, GJR-GARCH, TGARCH, 등을 사용 하고 있다.

ARCH 유도 과정 : 사례연구 hwp 파일 필요 공유

Ch15. Volatility, Implied Volatility, ARCH and GARCH

GARCH : Generalized **A**uto**R**egressive **C**onditional **H**eteroskedasticity

(일반 자기회귀 조건부 이분산성)

→ 현실에서는 ARCH 모형보다는 좀 더 일반화된 GARCH 모형 사용

ARCH 모형의 파생 것으로

어떤 시계열의 평균은 예측하지 못해도 변동성(분산)은 예측할 수 있는 경우라 할 수 있다.
금융시장에서 가격변수들을 예측할 때 널리 사용.

모형 GARCH(p, q)

$$\sigma_t^2 = \delta_0 + (\delta_1 \sigma_{t-1}^2 + \delta_2 \sigma_{t-2}^2 + \dots + \delta_p \sigma_{t-p}^2) + (\alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \dots + \alpha_q \epsilon_{t-q}^2)$$

일반적으로 사용되는 모형 GARCH(1, 1)

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

Ch15. Volatility, Implied Volatility, ARCH and GARCH

마르코프 연쇄(Markov chain) 이산시간 확률과정

마르코프 연쇄는 시간에 따른 계의 상태의 변화를 나타낸다.

매 시간마다 계는 상태를 바꾸거나 같은 상태를 유지한다. 상태의 변화를 전이라 한다.

마르코프 성질은 과거와 현재 상태가 주어졌을 때의 미래 상태의 조건부 확률 분포가 과거 상태와는 독립적으로 현재 상태에 의해서만 결정된다는 것을 뜻한다

<https://brilliant.org/wiki/markov-chains/>

A Markov chain is a mathematical system that experiences transitions from one state to another according to certain probabilistic rules.

The defining characteristic of a Markov chain is that no matter how the process arrived at its present state, the possible future states are fixed.

In other words, the probability of transitioning to any particular state is dependent solely on the current state and time elapsed.

The state space, or set of all possible states, can be anything: letters, numbers, weather conditions, baseball scores, or stock performances.

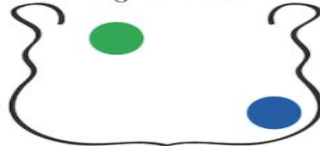
Stochastic Process

Random Variable



Possible States: ● ● ●

Bag of Balls



again for ball color, but it allows replacement each time a ball is drawn. Once again, the process, however, does satisfy the Markov property. Can you figure out why?

Markov Chain

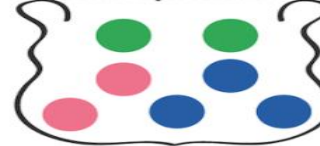
Random Variable

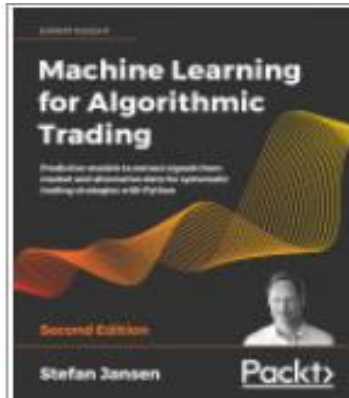


Possible States: ● ● ●

Bag of Balls

With replacement!





Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python

이후 page 내용은 위 교재에서 발췌한 내용과
추가 내용 보충한 것입니다. 학습용으로 참고 바랍니다.

파이썬 코드 다운로드. 기본내용

<https://github.com/PacktPublishing/Machine-Learning-for-Algorithmic-Trading-Second-Edition>

Ch1. Machine Learning for Trading: From Idea to Execution

- Changes in the **market microstructure**, such as the spread of electronic trading and the integration of markets across asset classes and geographies
- The development of investment strategies framed in terms of **risk-factor exposure**, as opposed to asset classes
- The revolutions in **computing power, data generation and management**, and **statistical methods**, including breakthroughs in deep learning
- The **outperformance of the pioneers** in algorithmic trading relative to human, discretionary investor

- From electronic to high-frequency trading
- Factor investing and **smart beta funds**
- Algorithmic pioneers **outperform** humans
- **ML-driven funds** attract \$1 trillion in AUM
- The emergence of quantamental funds
- Investments in strategic capabilities
- ML and alternative data
- Crowdsourcing trading algorithms

스마트 베타 ETF?

- 펀드 포트폴리오에 포함될 투자를 선택하기 위해 **규칙기반시스템**을 사용하는 **상장지수펀드** (ETF)
- 상장지수펀드 특정지수를 추종하는 펀드 유형 → 스마트 베타 ETF는 기존 ETF를 기반으로 하며 미리 결정된 재무지표를 기반으로 펀드의 보유 구성 요소를 조정하는 유형 펀드

Crowdsourcing trading algorithms

More recently, several algorithmic trading firms have begun to offer investment platforms that provide access to data and a programming environment to crowdsource risk factors that become part of an investment strategy or entire trading algorithms. Key examples include WorldQuant, Quantopian, and, most recently, Alpha Trading Labs (launched in 2018).

Machine Learning for Trading: From Idea to Execution

ETF(Exchanged Traded Fund)

Listing + index fund

- 공모 펀드 한계점 극복
- 기준가격 즉시 결정 효과(0)
- 주식처럼 매매 펀드
- 투자 : 분산투자 (간접적)

<https://www.investopedia.com/etfs-4427784>

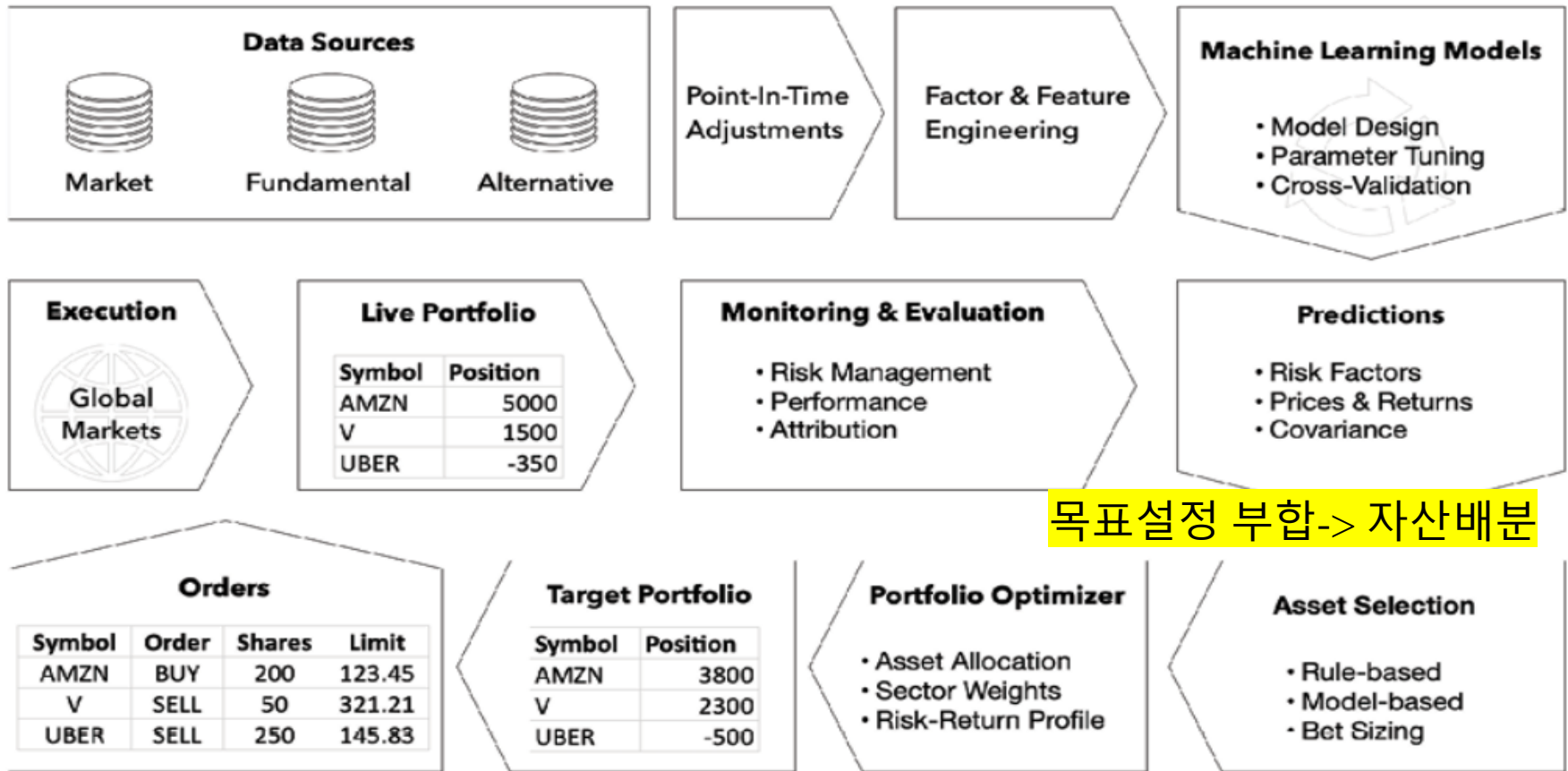
Table of Contents

- **1. Dollar-Cost Averaging**
- 2. Asset Allocation
- 3. Swing Trading
- 4. Sector Rotation
- 5. Short Selling
- 6. Betting on Seasonal Trends
- 7. Hedging
- The Bottom Line

Ch1. Machine Learning for Trading: From Idea to Execution

Designing and executing an ML-driven strategy

The ML4T Workflow



Machine Learning for Trading: From Idea to Execution

The evolution of algorithmic strategies

Quantitative strategies have evolved and become more sophisticated in three waves:

1. In the 1980s and 1990s, signals often emerged from **academic research** and used a single or very few inputs derived from market and fundamental data. AQR, one of the largest quantitative hedge funds today, was founded in 1998 to implement such strategies at scale. These signals are now largely commoditized and available as ETF, such as basic mean-reversion strategies.
2. In the 2000s, **factor-based investing** proliferated based on the pioneering work by Eugene Fama and Kenneth French and others. Funds used algorithms to identify assets exposed to risk factors like value or momentum to seek arbitrage opportunities. Redemptions during the early days of the financial crisis triggered the quant quake of August 2007, which cascaded through the factor-based fund industry. These strategies are now also available as long-only smart beta funds that tilt portfolios according to a given set of risk factors.
3. The third era is driven by investments in **ML capabilities and alternative data to generate** profitable signals for repeatable trading strategies. Factor decay is a major challenge: the excess returns from new anomalies have been shown to drop by a quarter from discovery to publication, and by over 50 percent after publication due to competition and crowding.

Machine Learning for Trading: From Idea to Execution

The evolution of algorithmic strategies

Today, traders pursue a range of different objectives when using algorithms to execute rules:

- Trade execution algorithms that aim to achieve favorable pricing
- Short-term trades that aim to profit from small price movements, for example, due to arbitrage (long&short fund)
- Behavioral strategies that aim to anticipate the behavior of other market participants
- Trading strategies based on absolute and relative price and return predictions

Use cases of ML for trading

- Data mining to identify patterns, extract features, and generate insights
- Supervised learning to generate risk factors or alphas and create trade ideas
- The aggregation of individual signals into a strategy
- The allocation of assets according to risk profiles learned by an algorithm
- The testing and evaluation of strategies, including through the use of synthetic data
- The interactive, automated refinement of a strategy using reinforcement learning

03 Alternative Data for Finance: Categories and Use Cases

The **alternative data** revolution

- **Volume:**
- **Variety**
- **Velocity**

Sources of **alternative data**

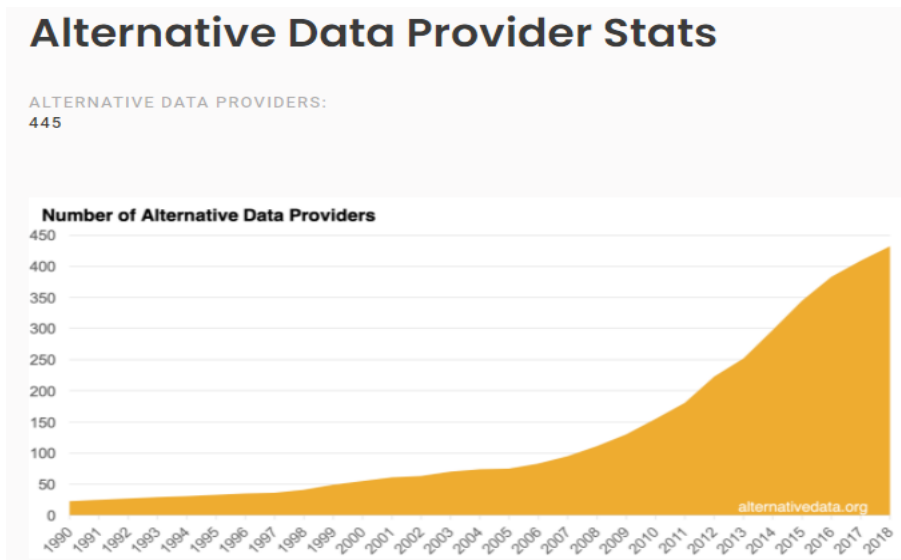
Alternative datasets are generated by many sources but can be classified at a high level as predominantly produced by:

- **Individuals** who post on social media, review products, or use search engines
- **Businesses** that record commercial transactions (in particular, credit card payments) or capture supply-chain activity as intermediaries
- **Sensors** that, among many other things, capture economic activity through images from satellites or security cameras, or through movement patterns such as cell phone towers

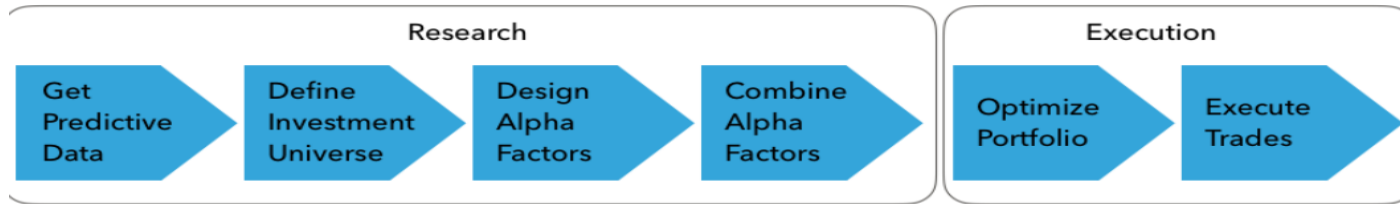
03 Alternative Data for Finance: Categories and Use Cases

<https://alternativedata.org/>

Alternative data refers to data used by investors to evaluate a company or investment that is not within their traditional data sources (financial statements, SEC filings, management presentations, press releases, etc.). Alternative data helps investors get more accurate, faster, or more granular insights and metrics into company performance than traditional data sources. Over the last 10 years, increases in computing power and personal device usage created massive growth in data generation. As a direct outcome, a large number of companies emerged to collect, clean, analyze, and interpret data and provide it as a product that could inform investment decisions (“Alternative Data Providers”). See growth in alternative data providers selling to institutional investors in Figure 1.



04 Financial Feature Engineering: How to research Alpha Factors



Index Fund 와 알파펀드

Index Fund에서 가장 큰 분화를 보인 것은 상장지수펀드(ETF, Exchange Traded Fund)임.

1990년대 초반 등장한 ETF는 2000년대 초반까지 정체상태를 보였지만, 2000년대 중반부터 자산군의 확대(채권-2002년, 상품-2004년)와 투자전략의 다양화(인버스/레버리지-2006년), Alternative Asset의 상품화(변동성-2009년) 등을 거치면서 급속도로 성장

ETF의 급격한 성장 배경은 대체로

- ① 저비용
- ② 투명성
- ③ 환금성에서 찾을 수 있음.

현물납입 원칙의 설정환매(in-Kind Creation/Redemption)방식이므로 거래비용이 최소화될 수 있으며(저비용), 포트폴리오를 실시간 공개하여 시장가격과 순자산가치의 비교를 통한 매매가 가능함(투명성).

거래소에 상장되어 실시간 매매가 가능하기 때문

국내에서는 개별 Index Fund보다는 ETF가 꾸준한 성장세를 기록하였음.

2002년 10월 14일에 상장한 삼성자산운용의 'KODEX200'은 최초 설정규모가 1,800억원 수준 이후 급속 성장 → 개별 ETF으로 자산규모가 가장 큰 ETF는 삼성자산운용의 'KODEX200'임.

ETF 시장은 최근 몇 년 동안 빠른 속도로 성장

지난 2016년 25조1000억원이었던 ETF 순자산총액은 지난해 52조를 기록하며 4년 만에 두 배로 성장했다.

2021년 11월 25일 기준으로 작년보다도 더 증가한 71조원 (세계 5위)

04 Financial Feature Engineering: How to research Alpha Factors

Index Fund 와 알파펀드

효율적 시장가설(EMH) 지지 Index Fund

- Vanguard Index Fund가 1976년 8월30일 출시
- 2008년 금융위기 이후 Active 퇴조와 Passive의 확장
- 2021.7.세계 2위 미국 뱅가드가 맞춤형 투자 포트폴리오 서비스를 제공하는 미 자산관리업체 저스트인베스트를 인수합병(M&A)→ 뱅가드 46년 역사상 처음

- **Vanguard Total Stock Market Index Fund(VTSAX)**

-최소 투자 : \$ 3,000,,비용 비율: 0.04%.

-Vanguard Total Stock Market Index Fund는 전체 주식 시장(VTSAX)에 투자

-중소형, 대형주 및 가치주를 포함한 미국 전체 주식 시장에 투자

- **뱅크드 토털 채권 시장 인덱스 펀드**

-회사채에 약 30%, 미국 국채에 약 70%를 투자하는 미국 투자 등급 채권

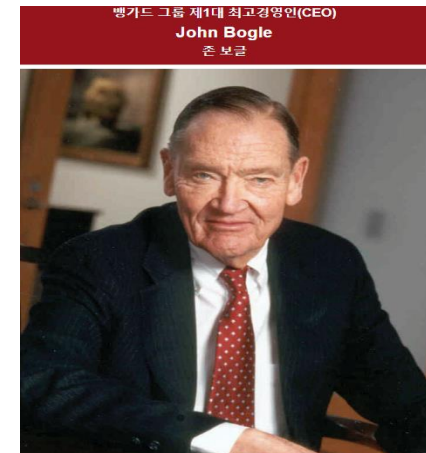
- **뱅크드 밸런스드 인덱스 펀드**

-60%가 주식에, 40%가 채권에 투자

- **뱅크드 선진시장 인덱스 펀드(VMGX)**

-미국 이외의 시장에 있는 대기업, 중소 기업에 투자

-FTSE Develop All Cap ex US 지수의 성과 추적



04 Financial Feature Engineering: How to research Alpha Factors

- **미국 INDEX FUND**

Wilshire Large Growth : 매출액성장률과 다른 성장률 지표로 750개의 대형주 평가

S&P 500 : 500개의 대형주, 가치주와 성장주

Wilshire Large Value : P/E 와 P/B가 가장 낮고 배당률이 가장 높은 750개의 대형주 평가

Wilshire Mid-Cap Growth : Wilshire Large Growth와 같은 방식으로 501위부터 1,250위까지의 중형주 평가

S&P 400 : 501위~900위까지의 중형주, 가치주와 성장주 모두

Wilshire Mid-Cap Value : Wilshire Large Growth와 같은 방식으로 501위부터 1,250위까지의 중형주 평가

Wilshire Small Growth : Wilshire Large Growth와 같은 방식으로 751위부터 2,500위까지의 소형주 평가

Russell 2000 : 1,001위부터 3,000위 까지의 소형주, 가치주와 성장주 모두

국제 지수 : MSCI The World

미국을 포함해 모든 선진국과 산업의 60% 포함

MSCI EAFE : 미국을 제외한 유럽, 호주, 극동의 20개국 시가총액의 60% 포함

MSCI Emerging Markets : 내부 지침에 의해 신흥시장으로 구분되는 국가에 MSCI 기준 적용

채권 지수

Lehman Bros. Long-Term Govt/Corp : 액면 1억 달러 이상, 만기 10년 이상의 재무부, 정부기관, 기업의 채권

Lehman Bros. Interm-Term Govt/Corp : Lehman Bros. Long-Term Govt/Corp 와 같은 기준에 만기는 1년 이상 10년 미만의 채권

04 Financial Feature Engineering: How to research Alpha Factors

Index Fund 와 알파펀드

John Bogle, “The Index Mutual Fund: 40 Years of Growth, Change, and Challenge”, Journal of Financial Analyst, Jan/Feb 2016).

‘Smart beta’

risk factor에 대한 overweight 또는 underweight을 통해 시장 대비 초과수익을 추구하는 투자 전략

→ 특정 Factor에 노출된 Alpha 전략 일환 정의

→ (Josh Barrickman, “Why smart beta can't win the indexing race”, Vanguard Blog for Advisor, Feb. 2015)

04 Financial Feature Engineering: How to research Alpha Factors

Index Fund 와 알파펀드

Martijn Cremers, Miguel Ferreira, Pedro Matos, Laura Starks,

"Indexing and Active Fund Management: International Evidence", Journal of Financial Economics, forthcoming, 2015

Moreover, the average alpha generated by active management is higher in countries with more explicit indexing and lower in countries with more closet indexing. Overall, our evidence suggests that explicit indexing improves competition in the mutual fund industry.

Active Shares' : 펀드내 개별주식의 비중과 벤치마크 내 해당 주식의 비중을 비교하는 개념
순수 인덱스펀드는 '0'의 'Active Shares'를 갖지만 ,Active 펀드일수록 수치가 높아짐. '

Active Shares'는 운용자(manager)의 능력과 개별종목에 대한 확신, 이상현상과 같은 투자기회의 활용 정도를 반영하는 지표임.

'Active Shares'의 비율이 높을수록 감수할 위험의 정도와 함께 기대수익률도 상승하며, 이에 기반하여 펀드수수료가 책정될 수 있음.
Cremers & Petajisto('Active Shares' 개념 처음 제시) 'Active Shares' 비중 60% 이하를 'Closet Indexing'으로 분류하였음.

Active 펀드 운용자는 펀드내 구성종목 중 최대 40%에 해당하는 종목으로 미래의 초과성과(Alpha)를 창출해야 하며, 역으로 펀드 구성 종목 중 60%에 해당하는 종목은 벤치마크 성과(Beta)를 달성하는데 한정한다는 것임.

결국 Active 운용을 표방하지만 실제 운용스타일은 Passive 운용에 가깝게 유지하면서 상대적으로 높은 운용보수를 수취하고 있음.
해당 논문에서 각국의 펀드시장에 대한 실증분석(2010년말 기준 32개국 24,492개 펀드(순자산규모 9.8조원) 대상)을 통해 평균적으로 58%가 Active 펀드이며, 명시적인(explicit) Indexing 비율은 22%이고 Closet Indexing의 비중은 20% 수준인 것으로 나타났음. 명시적인(explicit) Indexing이 발달할수록 Active 펀드의 비용이 낮아지는 것이 관찰되었음. 또한 Index 펀드와 ETF의 비중이 높을수록 Active 펀드의 성과가 개선되는 반면, Closet Indexing 비중이 높을수록 Active 펀드의 성과가 상대적으로 낮은 것으로 분석되었음.

Closet Indexing은 Active Fund의 Passive 운용전략 채용이라는 투자전략의 모호함과 함께 Passive Fund에 비해 상대적으로 높은 보수를 수취하고 있다는 점에서 투자자 보호차원의 이슈를 제기할 수 있음.

04 Financial Feature Engineering: How to research Alpha Factors

Index Fund 와 알파펀드

The portion of an asset's return that is **not explained by exposure to this benchmark** is called **alpha**, **and hence the signals** that aim to produce such uncorrelated returns are also called **alpha factors**.

Alpha factors are transformations of raw data **that aim to predict asset price movements**.

They are designed to **capture risks that drive asset returns**.

A factor may combine one or several inputs, but **outputs a single value** for each asset, every time the strategy evaluates the factor to obtain a signal.

Trade decisions may rely on **relative factor values across assets** or **patterns** for a single asset.

The **design, evaluation, and combination of alpha factors** are critical steps during the research phase of the algorithmic trading strategy workflow

04 Financial Feature Engineering: How to research Alpha Factors

Index Fund 와 알파펀드

Alphalens facilitates the analysis of the predictive power of alpha factors concerning the:

- **Correlation** of the signals with subsequent returns
- **Profitability** of an equal or factor-weighted portfolio based on a (subset of) the signals
- **Turnover of factors** to indicate the potential trading costs
- **Factor performance** during specific events
- **Breakdowns of the preceding** by sector

05 Portfolio Optimization and Performance Evaluation

Kalman filter?

-1960년대 초 루돌프 칼만이 개발한 알고리즘, NASA의 아폴로 프로젝트에서 네비게이션 개발 시에 사용
-관측데이터로 부터 위치를 예측하는 방법에 자주 등장→ 현재는 GPS, 날씨 예측, 주식 예측 등 다양한 예제에서 널리 사용

→ Kalman filter는 system을 선형화하여 해석하려는 방법
(가정)

- 1) 모션 모델과 측정 모델이 linear할 경우
- 2) 모션 모델과 측정 모델이 Gaussian 분포를 따를 경우

모션 모델

로봇이 현재위치에서 모션 입력을 입력받아서 움직였을 때의 확률 모델

측정 모델

로봇이 현재위치에서 자신이 가진 센서를 이용해서 자신이 어디에 위치해 있는지를 측정했을 때의 확률 모델
칼만 필터는 상태 예측(state prediction)과 측정 업데이트(measurement update)를 반복적으로 수행하며 로봇의 현재 위치를 계산한다.

1차원의 경우

상태 예측 단계

이전 측정 업데이트에서 계산한 확률 분포와 로봇 모션 입력의 확률 분포를 이용해 현재 상태의 분포를 예측
→ 이 때 확률분포의 평균은 간단히 두 평균을 더한 것이고 확률분포의 분산은 두 분산을 더한 것이 됩니다. 이는 1차원의 경우를 예로 들고 있기 때문에 간단한 합이 되지만 뒤에서 다룰 다차원의 경우에는 좀더 복잡한 수식으로 상태 예측이 수행

측정 업데이트

상태 예측단계에서 예측된 현재 로봇 위치에 대한 확률분포와 현재 로봇의 위치에서 측정한 관찰값의 확률 분포를 이용하여 사후 확률분포를 업데이트 하는 방식으로 수행

05 Portfolio Optimization and Performance Evaluation

Information Ratio(IR)

Information Coefficient (IC)

Information Ratio(IR) = α / tracking error

→ 위험조정초과성과

Information coefficient (IC) : 예측 주식수익률과 실제수익률과 같은 두 개의 랜덤 변수 간의 선형 관계를 측정할 수 있다는 점에서 상관관계와 유사

information coefficient (IC) is a measure of the merit of a predicted value. In [finance](#), the information coefficient is used as a performance metric for the predictive skill of a [financial analyst](#).^[1] The information coefficient is similar to [correlation](#) in that it can be seen to measure the linear relationship between two [random variables](#), e.g. predicted stock [returns](#) and the actualized returns. The information coefficient ranges from 0 to 1, with 0 denoting no linear relationship between predictions and actual values (poor forecasting skills) and 1 denoting a perfect linear relationship (good forecasting skills)

Measuring Investment Skill Using The Effective Information Coefficient

$$IR = IC * Breadth$$

- IC = correlation of your return forecasts and outcomes
- Breadth = number of independent “bets” taken per unit time

$$IR = IC * TC * Breadth$$

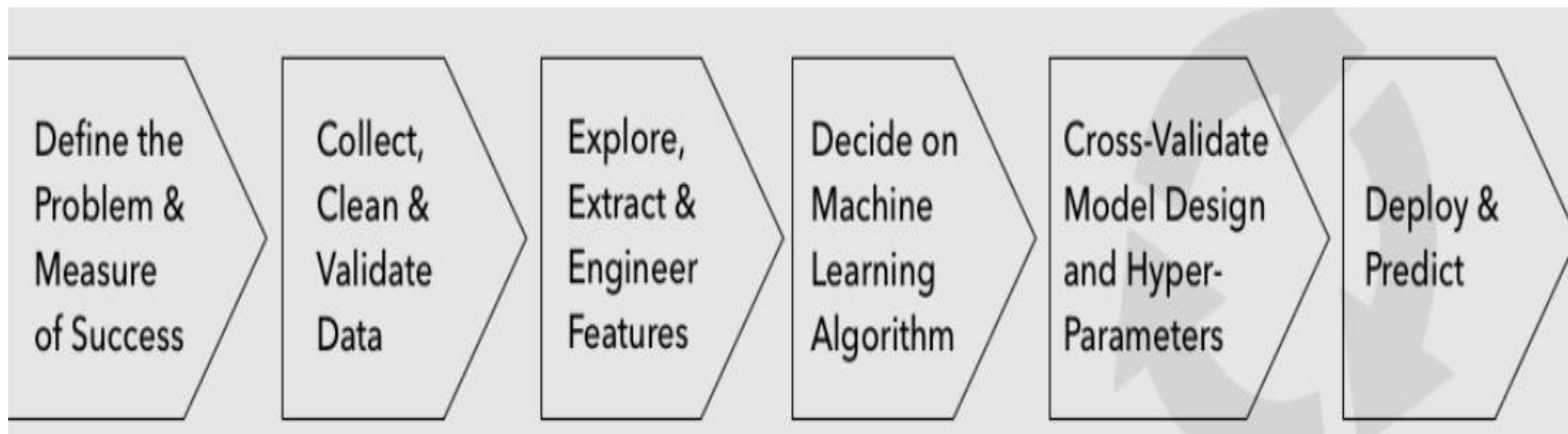
TC = the efficiency of your portfolio construction ($TC < 1$)

Information Coefficient as a Performance Measure of Stock Selection Models Feng Zhang, Ruite Guo and Honggao Cao1 Wells Fargo & Company October, 2020

06 The Machine Learning Process

- Linear models for the regression and classification of cross-section, time series, and panel data
- Generalized additive models, including nonlinear tree-based models, such as decision trees
- Ensemble models, including random forest and gradient-boosting machines
- Unsupervised linear and nonlinear methods for dimensionality reduction and clustering
- Neural network models, including recurrent and convolutional architectures
- Reinforcement learning models

The machine learning workflow



06 The Machine Learning Process

- How supervised and unsupervised learning from data works
- Training and evaluating supervised learning models for regression and classification tasks
- How the bias-variance trade-off impacts predictive performance
- How to diagnose and address prediction errors due to overfitting
- Using cross-validation to optimize hyperparameters with a focus on time-series data
- Why financial data requires additional attention when testing out-of-sample

06 The Machine Learning Process

How machine learning from data works

• **Arthru Samuel** (1959): Machine Learning is a field of study that gives computers the ability to learn **without being explicitly programmed**

Tom Mitchell (1988): A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on t , as measured by P , improves with experience E
→ 컴퓨터가 작업(T)을 하는데 있어 **경험(E)로부터 학습하여 성능에 대한 측정(P)을 개선시키는 분야**를 기계학습(ML)이라고 정의

(체커 게임 사례)

• → 컴퓨터는 체커게임에서 수많은 게임(T)을 통해서 경험(E)으로부터 승리할 확률(P)을 향상시켰기에 기계학습에 대한 대표적인 예시

7. Linear Models: From Risk Factors to Return Forecasts

Linear Models – From Risk Factors to Return Forecasts

The family of **linear models** represents one of the most useful hypothesis classes.

Many learning algorithms that are widely applied in algorithmic trading rely on linear predictors because **they can be efficiently trained, are relatively robust to noisy financial data, and have strong links to the theory of finance.**

Linear predictors are also intuitive, easy to interpret, and often fit the data reasonably well or at least provide a good baseline.

Linear regression has been known for over 200 years, since **Legendre and Gauss** applied it to astronomy and began to analyze its statistical properties.

Numerous extensions have since adapted the linear regression model and the baseline **ordinary least squares (OLS)** method to learn its parameters:

- **Generalized linear models (GLM)** expand the scope of applications by allowing for response variables that imply an error distribution other than the normal distribution...

covers the following topics:

- How **linear regression works** and which **assumptions** it makes
- Training and diagnosing linear regression models
- Using linear regression to **predict** stock returns
- Use **regularization** to improve the **predictive performance**
- How **logistic regression** works
- Converting a regression into a **classification problem**

7. Linear Models: From Risk Factors to Return Forecasts

Linear Models – From Risk Factors to Return Forecasts

Many learning algorithms that are widely applied in algorithmic trading rely on linear predictors because they can be efficiently trained, are relatively robust to noisy financial data, and have strong links to the theory of finance. Linear predictors are also intuitive, easy to interpret, and often fit the data reasonably well or at least provide a good baseline.

Linear regression has been known for over 200 years, since Legendre and Gauss applied it to astronomy and began to analyze its statistical properties. Numerous extensions have since adapted the linear regression model and the baseline **ordinary least squares (OLS)** method to learn its parameters:

Legendre and Gauss → Gauss-Legendre Quadrature

→ 가우스 쿼드러처. 가우스 구분구적법. 가우시안 쿼드러처. 가우시안 구분구적법

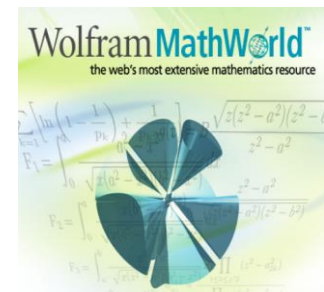
→ 가우스 르장드르 알고리즘 인 알고리즘의 숫자를 계산하는 π . 25번의 반복으로 4,500만 개의 올바른 자릿수 π 를 생성하는 등 빠르게 수렴하는 것으로 유명

단점은 컴퓨터 메모리 집약적이므로 때때로 Machin과 유사한 공식이 대신 사용된다는 것.

이 방법은 곱셈 및 제곱근에 대한 현대 알고리즘과 결합된 Carl Friedrich Gauss (1777–1855) 및 Adrien-Marie Legendre (1752–1833)의 개별 작업을 기반

<https://mathworld.wolfram.com/Legendre-GaussQuadrature.html>

<https://mathworld.wolfram.com>



7. Linear Models: From Risk Factors to Return Forecasts

The baseline model – multiple linear regression

$$\mathbf{X}^T = [x_1, \dots, x_p], \text{ and the error } \epsilon:$$

input vector

$$y = f(\mathbf{x}) + \epsilon = \beta_0 + \beta_1 x_1 \dots + \beta_p x_p + \epsilon = \beta_0 + \sum_{j=1}^p \beta_j x_j + \epsilon$$

```
import statsmodels.api as sm
X_ols = sm.add_constant(X)
model = sm.OLS(y, X_ols).fit()
model.summary()
```

7. Linear Models: From Risk Factors to Return Forecasts

How to build a linear factor model

Algorithmic trading strategies use factor models to quantify the relationship between the return of an asset and the sources of risk that are the main drivers of these returns. Each factor risk carries a premium, and the total asset return can be expected to correspond to a weighted average of these risk premia.

7. Linear Models: From Risk Factors to Return Forecasts

```
import statsmodels.api as sm
X_ols = sm.add_constant(X)
model = sm.OLS(y, X_ols).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	Y	R-squared:	0.791
Model:	OLS	Adj. R-squared:	0.790
Method:	Least Squares	F-statistic:	1176.
Date:	Thu, 14 Nov 2019	Prob (F-statistic):	4.33e-212
Time:	18:58:15	Log-Likelihood:	-3309.2
No. Observations:	625	AIC:	6624.
Df Residuals:	622	BIC:	6638.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	53.2923	1.934	27.561	0.000	49.495	57.089
X_1	0.9904	0.064	15.390	0.000	0.864	1.117
X_2	2.9600	0.064	45.996	0.000	2.834	3.086

Omnibus:	0.267	Durbin-Watson:	2.148
Prob(Omnibus):	0.875	Jarque-Bera (JB):	0.149
Skew:	0.014	Prob(JB):	0.928
Kurtosis:	3.071	Cond. No.	30.0

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

- Durbin Watson(DW) test : 오차항에 자기상관이 있는지 여부 검정하기 위해 사용
→ 2에 가까울수록 → 오차항의 자기상관이 없음(독립성 가정 만족)
- Jarque-Bera (JB) -> Normality test
→ JB가 0보다 충분히 큰 경우, 즉 오른쪽 꼬리의 값이 나올 때 귀무가설을 기각한다
- AIC : 다중공선성, 설명변수를 가지는 회귀분석의 경우 설명변수들 사이의 독립성 등의 가정을 만족시키기 어렵고, 또한 설명변수의 증가는 모형의 결정계수 왜곡

7. Linear Models: From Risk Factors to Return Forecasts

OLS Regression Results

```
=====
                        OLS Regression Results
=====
Dep. Variable:          GRADE      R-squared:                0.416
Model:                  OLS        Adj. R-squared:            0.353
Method:                 Least Squares    F-statistic:              6.646
Date:                  Fri, 12 Nov 2021    Prob (F-statistic):      0.00157
Time:                  23:41:16      Log-Likelihood:          -12.978
No. Observations:      32           AIC:                     33.96
Df Residuals:          28           BIC:                     39.82
Df Model:              3
Covariance Type:       nonrobust
=====
                        coef      std err          t      P>|t|      [0.025      0.975]
-----
GPA                0.4639      0.162        2.864      0.008      0.132      0.796
TUCE               0.0105      0.019        0.539      0.594     -0.029      0.050
PSI                0.3786      0.139        2.720      0.011      0.093      0.664
const             -1.4980      0.524       -2.859      0.008     -2.571     -0.425
=====
Omnibus:            0.176    Durbin-Watson:           2.346
Prob(Omnibus):      0.916    Jarque-Bera (JB):        0.167
Skew:              0.141    Prob(JB):                0.920
Kurtosis:          2.786    Cond. No.                176.
=====
```

7. Linear Models: From Risk Factors to Return Forecasts

Generalized Linear Models

Generalized Linear Model Regression Results

```
=====
Dep. Variable:          YES      No. Observations:          32
Model:                  GLM      Df Residuals:              24
Model Family:           Gamma    Df Model:                  7
Link Function:          inverse_power  Scale:                0.0035843
Method:                  IRLS     Log-Likelihood:         -83.017
Date:                   Fri, 12 Nov 2021  Deviance:             0.087389
Time:                   23:41:12    Pearson chi2:           0.0860
No. Iterations:         6          Pseudo R-squ. (CS):      0.9800
Covariance Type:        nonrobust
=====
```

	coef	std err	z	P> z	[0.025	0.975]
const	-0.0178	0.011	-1.548	0.122	-0.040	0.005
COUTAX	4.962e-05	1.62e-05	3.060	0.002	1.78e-05	8.14e-05
UNEMPF	0.0020	0.001	3.824	0.000	0.001	0.003
MOR	-7.181e-05	2.71e-05	-2.648	0.008	-0.000	-1.87e-05
ACT	0.0001	4.06e-05	2.757	0.006	3.23e-05	0.000
GDP	-1.468e-07	1.24e-07	-1.187	0.235	-3.89e-07	9.56e-08
AGE	-0.0005	0.000	-2.159	0.031	-0.001	-4.78e-05
COUTAX_FEMALEUNEMP	-2.427e-06	7.46e-07	-3.253	0.001	-3.89e-06	-9.65e-07

```
=====
```

7. Linear Models: From Risk Factors to Return Forecasts

Generalized Estimating Equations

Generalized Estimating Equations estimate generalized linear models for panel, cluster or repeated measures data when the observations are possibly correlated within a cluster but uncorrelated across clusters. It supports estimation of the same one-parameter exponential families as Generalized Linear models (*GLM*)

GEE Regression Results

```
=====
Dep. Variable:          y      No. Observations:          236
Model:                  GEE    No. clusters:              59
Method:                 Generalized  Min. cluster size:      4
                        Estimating Equations  Max. cluster size:      4
Family:                 Poisson  Mean cluster size:      4.0
Dependence structure:   Exchangeable  Num. iterations:        2
Date:                  Fri, 12 Nov 2021  Scale:                1.000
Covariance type:        robust  Time:                   23:43:15
=====
```

	coef	std err	z	P> z	[0.025	0.975]
Intercept	0.5730	0.361	1.589	0.112	-0.134	1.280
trt[T.progabide]	-0.1519	0.171	-0.888	0.375	-0.487	0.183
age	0.0223	0.011	1.960	0.050	2.11e-06	0.045
base	0.0226	0.001	18.451	0.000	0.020	0.025

```
=====
Skew:                  3.7823  Kurtosis:                28.6672
Centered skew:         2.7597  Centered kurtosis:      21.9865
=====
```

7. Linear Models: From Risk Factors to Return Forecasts

Generalized Additive Models (GAM)

Generalized Additive Models allow for **penalized** estimation of smooth terms in generalized linear models.

- 설명변수 각 x_j 에 대해 별도로 f_j 를 계산하고 그 다음에 이들의 **기여를 모두 더하기 때문에 가법모델**이라 함.
- 일반화 가법모형(GAM)은 생물학, 의학, 환경학 등 여러 분야에서 접할 수 있는 **복잡한 비선형관계를 분석할** 수 있게 해주는 다재 다능한 모형
- 통계학적인 모형을 만들 때 모형의 유연성(flexibility)과 해석가능성(interpretability)은 서로 상충하는 관계(trade-off)
- 선형회귀와 같은 단순한 모형은 사용하기 쉽고 해석하기 쉬우며 인수의 뜻도 해석하기 쉽다.
- 선형관계로 설명할 수 없는 현상을 설명하려면 보다 복잡한 모형이 필요하다.
- 신경망과 같은 기계학습 모형은 복잡한 관계를 예측하는 능력은 뛰어나지만 많은 데이터를 필요로 하고 해석하기 어렵고 모형의 결과로부터 추론하는 것이 매우 어렵다.
- 복잡한 비선형관계를 적합시킬 수 있으며 매우 좋은 예측을 할 수 있는 반면 통계적 추론이 가능할 뿐만 아니라 모형의 구조를 이해하고 설명할 수 있으며 예측에 대한 설명도 가능

7. Linear Models: From Risk Factors to Return Forecasts

다중 일반화가법모형 (Multiple GAMs)

GAM의 설명변수로 여러 개의 변수를 선택할 수 있으며 연속형 변수인 경우 곡선형태의 비선형효과 또는 선형효과를 선택할 수 있으며 범주형 변수도 선택 가능하며 범주형 변수의 각각의 범주에 따라 서로 다른 비선형 효과를 선택

모형의 평활항

모수 항 다음에는 평활항에 대한 부분. 평활항의 회귀계수는 출력되지 않는데 그 이유는 각 평활항마다 기저함수의 갯수에 따라 회귀계수가 여러개 존재하기 때문이다.

회귀계수 대신 첫번째 열에 *edf*가 출력되는데 이는 유효자유도(effective degrees of freedom)이다.

이 값은 평활항의 복잡성을 나타내준다.

*edf*가 1인 것은 직선을 뜻하며 *edf*가 2인 것은 제곱항을 뜻한다. *edf*가 클수록 곡선은 더욱 복잡해진다.

7. Linear Models: From Risk Factors to Return Forecasts

Robust Linear Models

- **일반 회귀분석**: 예측식을 구성하는 독립변수들의 예측계수(회귀계수)를 구할 때 잔차의 제곱의 합이 최소가 되도록 최소 제곱법(Method of Ordinary Least Squares, OLS)을 적용하는 방법
 - **Robust Linear Models**: 잔차의 절대값의 합이 최소가 되도록 계수를 추정하는 방식
- ➔ 이는 임의적으로 이상치(Outlier)제거를 하는 것이 아니라 강건(Robust)회귀분석 방법론을 통해 모형을 추정

Robust linear Model Regression Results

```
=====
Dep. Variable:          y      No. Observations:          21
Model:                  RLM      Df Residuals:             17
Method:                 IRLS      Df Model:                 3
Norm:                   HuberT
Scale Est.:             mad
Cov Type:               H1
Date:                   Fri, 12 Nov 2021
Time:                   23:30:43
No. Iterations:         19
=====
```

	coef	std err	z	P> z	[0.025	0.975]
var_0	-41.0265	9.792	-4.190	0.000	-60.218	-21.835
var_1	0.8294	0.111	7.472	0.000	0.612	1.047
var_2	0.9261	0.303	3.057	0.002	0.332	1.520
var_3	-0.1278	0.129	-0.994	0.320	-0.380	0.124

```
=====
```

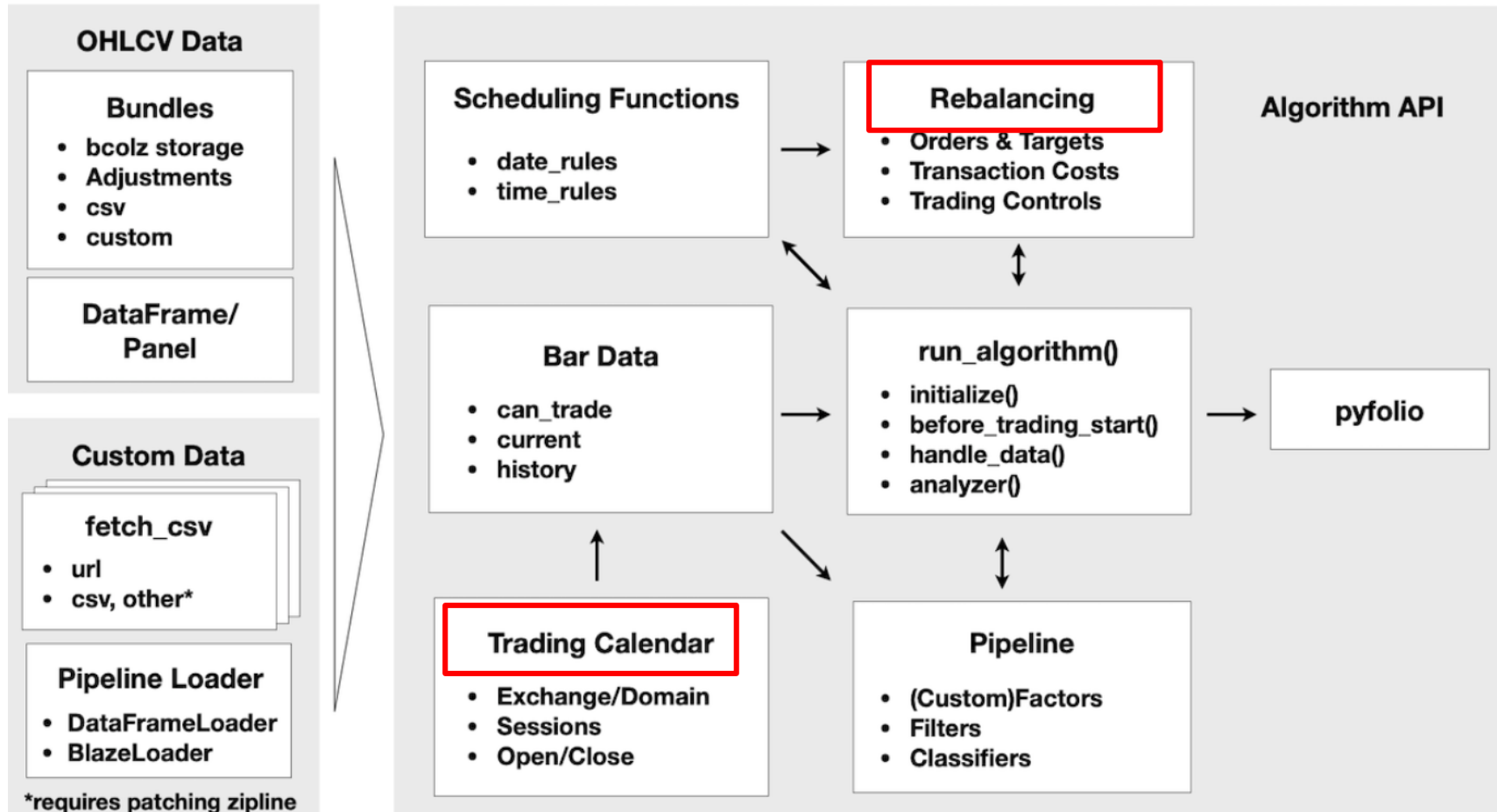
7. Linear Models: From Risk Factors to Return Forecasts

Comparing OLS and RLM

https://www.statsmodels.org/stable/examples/notebooks/generated/robust_models_0.html

08 The ML4T Workflow: From Model to Strategy Backtesting

Zipline Architecture



09 Time Series Models for Volatility Forecasts and Statistical Arbitrage

From inference to prediction

The model also assumes a random error that allows for each observation to deviate from the expected linear relationship.

The reasons that the model does not perfectly describe the relationship between inputs and output in a deterministic way include, for example, missing variables, measurement, or data collection issues.

If we want to draw statistical conclusions about the true (but not observed) linear relationship in the population based on the regression parameters estimated from the sample, we need to add assumptions about the statistical nature of these errors.

The baseline regression model makes the strong assumption that the distribution of the errors is identical across observations. It also assumes that errors are independent of each other—in other words, knowing one error does not help to forecast...

09 Time Series Models for Volatility Forecasts and Statistical Arbitrage

In particular, it covers:

- How to use **time-series analysis** to prepare and inform the modeling process
- Estimating and diagnosing **univariate autoregressive and moving-average models**
- Building autoregressive conditional heteroskedasticity (**ARCH**) models to **predict volatility**
- How to build **multivariate vector autoregressive** models
- Using cointegration to develop a **pairs trading strategy**

09 Time Series Models for Volatility Forecasts and Statistical Arbitrage

The baseline model – multiple linear regression

How to formulate the model

The **multiple regression model** defines a linear functional relationship between one **continuous outcome variable** and **p input variables** that can be of any type but may require preprocessing.

Multivariate regression, in contrast, refers to the regression of multiple outputs on multiple input variables.

In the population, the linear regression model has the following form for a single instance of the output y , an input vector

$$\mathbf{X}^T = [x_1, \dots, x_p], \text{ and the error } \epsilon :$$

$$y = f(\mathbf{x}) + \epsilon = \beta_0 + \beta_1 x_1 \dots + \beta_p x_p + \epsilon = \beta_0 + \sum_{j=1}^p \beta_j x_j + \epsilon$$

09 Time Series Models for Volatility Forecasts and Statistical Arbitrage

Regularizing linear regression using shrinkage

회귀분석(Regression Analysis)은 설명변수를 설정 · 활용하여 종속변수를 설명하는 분석 방법

- 설명변수가 많은 경우 불필요한 부분까지 설명하는 과적합 문제가 발생할 수 있음
- 설명변수 축소방법(Shrinkage Methods)을 이용한 회귀분석은 과적합 문제를 해결할 수 있는 장점이 있어 설명변수가 많은 경우 주로 활용함
- 회귀분석이 RSS(Residual Sum of Square)을 최소화하는 β 를 추정한다면,
- 설명변수 축소방법(Shrinkage Methods)을 이용한 회귀분석은 RSS에 Penalty 조건을 추가하여 "RSS + Penalty"를 최소화하는 β 추정한 것(L1, L2, L1+L2)

09 Time Series Models for Volatility Forecasts and Statistical Arbitrage

How to predict returns with linear regression

statsmodels

<https://www.statsmodels.org/stable/index.html>

Regression and Linear Models¶

- Linear Regression
- Generalized Linear Models
- Generalized Estimating Equations
- Generalized Additive Models (GAM)
- Robust Linear Models
- Linear Mixed Effects Models
- Regression with Discrete Dependent Variable
- Generalized Linear Mixed Effects Models
- ANOVA
- Other Models othermod

Time Series Analysis¶

- Time Series analysis tsa
- Time Series Analysis by State Space Methods statespace
- Vector Autoregressions tsa.vector_ar

Other Models¶

- Methods for Survival and Duration Analysis
- Nonparametric Methods nonparametric
- Generalized Method of Moments gmm
- Other Models miscmodels
- Multivariate Statistics multivariate

Statistics and Tools¶

- Statistics stats
- Contingency tables
- Multiple Imputation with Chained Equations
- Empirical Likelihood emplike
- Distributions
- Graphics
- Input-Output iolib
- Tools
- Working with Large Data Sets
- Optimization

09 Time Series Models for Volatility Forecasts and Statistical Arbitrage

a **stationary time series** are independent of the period—that is, they don't change over time. Thus, **stationarity** implies that **a time series does not** have a trend or seasonal effects.

- 시계열 분석
- 장점: 예측 값 추정
- 단점: 변수간 이론적 관계를 고려하지 못함.

09 Time Series Models for Volatility Forecasts and Statistical Arbitrage

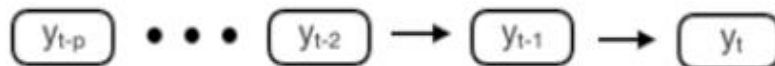
Univariate time-series models

Multiple linear-regression models expressed the variable of interest as a linear combination of the inputs, plus a random disturbance. In contrast, univariate time-series models relate the current value of the time series to a linear combination of lagged values of the series, current noise, and possibly past noise terms. While exponential smoothing models are based on a description of the trend and seasonality in the data, **ARIMA models aim to describe the autocorrelations in the data**. ARIMA(p, d, q) models require stationarity and leverage two building blocks:

- **Autoregressive (AR)** terms consisting of p lagged values of the time series
- **Moving average (MA)** terms that contain q lagged disturbances

Univariate Time Series

ARMA Models

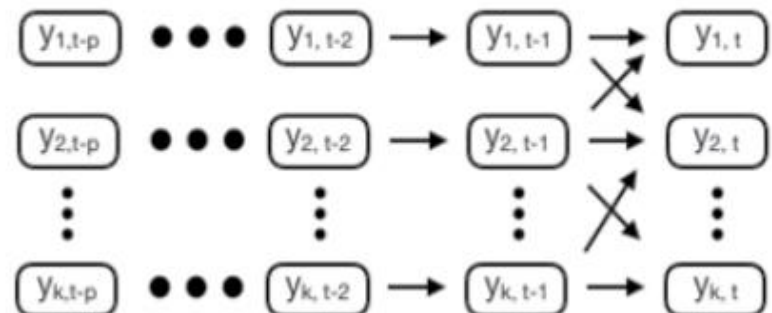


with exogenous variables



Multivariate Time Series

Vector Autoregressive (VAR) Models



09 Time Series Models for Volatility Forecasts and Statistical Arbitrage

Statistical arbitrage with cointegration

Statistical arbitrage refers to strategies that employ some statistical model or method to take advantage of what appears to be relative mispricing of assets, while maintaining a level of market neutrality.

Pairs trading is a conceptually straightforward strategy that has been employed by algorithmic traders since at least the mid-eighties (Gatev, Goetzmann, and Rouwenhorst 2006).

The goal is to find two assets whose prices have historically moved together, track the spread (the difference between their prices), and, once the spread widens, **buy the loser that has dropped below the common trend and short the winner**.

If the relationship persists, the long and/or the short leg will deliver profits as prices converge and the positions are closed.

This approach extends to a multivariate context by forming baskets from multiple securities and trading one asset against a basket of two baskets against each other.

In practice, the strategy...

09 Time Series Models for Volatility Forecasts and Statistical Arbitrage

A time series is a sequence of values separated by discrete intervals that are typically even spaced (except for missing values). A time series is often modeled as a **stochastic process** consisting of a collection of random variables,

* 내용 추가 : Time-series

시계열자료 (Time-series)

시간의 순서에 따라 기록된 데이터

이동평균 기법(Moving Average method) : 미래 예측값은 지정된 기간의 과거 데이터를 평균한 값과 동일하다. → 산술평균

단순 지수평활(Simple Exponential Smoothing) : 이전 시계열에 대해 가중을 더하는 방법으로 오래된 관측값 일수록 가중이 지수적으로 작아지는 방식으로 미래 예측값을 계산한다. 이 방법은 추세나 계절성이 없는 시계열을 예측할 때 유용하다. 다음에 나오는 예측방법에서는 단순 지수평활이 수준식으로 불리게 될 것이다.

$$\hat{y}_{T+h|T} = \alpha y_T + \alpha(1 - \alpha)y_{T-1} + \alpha(1 - \alpha)^2 y_{T-2} + \dots$$

홀트의 선형 추세 기법(Holt's Linear Trend Method)

: 단순 지수평활에 추세를 반영한 모델

예측함수가 더이상 평평하지 않고 추세를 가진다.

홀트-윈터스의 계절성 기법(Holt-Winters seasonal method) : 시계열의 계절성을 잡아내기 위해

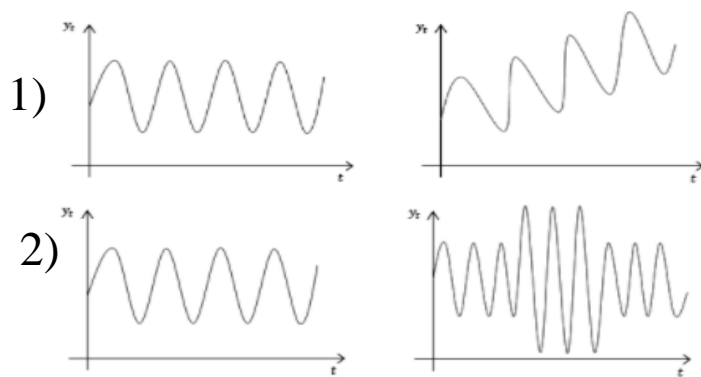
홀트의 선형 추세 기법에 계절성을 반영한 모델

* 내용 추가 : Time-series

시계열분석 : 정상성 자료 이용해야 함.

정상성(Stationarity)

시계열 자료(y_t)의 평균과 분산이 t 시점에 상관없이 동일할 때와 y_t 와 y_{t-h} 시점의 자기 상관은 시차(h)에만 의존하며 시점($t/t-h$)에는 의존하지 않는 경우
→ 만약 추세나, 계절성이 있으면 정상성 시계열이 아니다.



정상성

비정상성

데이터가 정상성을 가지지 않으면 분석이 어렵기 때문에 정상성을 갖도록 데이터 처리 과정 필요

- 1) 평균이 일정하지 않으면 → 차분을 취하고
- 2) 분산이 일정하지 않으면 → log변환

비정상성 (Non-stationary) 시계열 : 분산이 시점 t 에 의존

- 1980년대 이전 : 시계열분석시 시계열변수들이 stationary하다는 가정을 설정하여 연구
- Nelson & Plosser(1982) : 단위근(확률적 추세)을 가지는 Non-stationary 시계열에 의해 보다 나은 적합한 모형화

* 내용 추가 : Time-series

백색잡음(White Noise)

정상시계열의 사례로 백색잡음은 시점에 상관없이 평균이 0이고 분산이 σ^2 인 시계열자료

- ϵ (백색잡음)는 확률적 오차항으로 평균이 일정(보통 0으로 가정)하고 분산 σ^2 을 가지는 무상관인 확률변수로 정의, 자기상관이 없는 것으로 가정
- 백색잡음의 평균과 분산은 각각 0, σ^2 이므로 시점 t 에 영향을 받지 않는다.
- 자기 공분산시점 t 와 무관하므로 정상성을 만족하는 정상(stationary) 시계열

$$y_t = \epsilon_t WN(0, \sigma^2)$$

$$Cov(\epsilon_t, \epsilon_{t-h}) = \begin{cases} \sigma^2, h = 0 \\ 0, h \neq 0 \end{cases}$$

* 내용 추가 : Time-series

확률보행(Random Walk)

- 확률보행은 임의의 방향으로 향하는 연속적인 걸음을 나타낸다는 의미로 예측 불가능한 변동이 발생하는 것
- 미래 이동을 예측할 수 없기 때문에 확률보행 모델에서 예측값은 마지막 예측값과 같다고 가정

$$y_t = y_{t-1} + \epsilon_t, \epsilon_t \text{ iid}(0, \sigma^2)$$

- 확률보행모델의 오차항은 백색잡음이 아니다.
- 확률보행과정은 분산이 시점 t 에 의존하므로 비정상(non-stationary) 시계열

$$E(y_t) = E(\epsilon_1 + \epsilon_2 + \cdots + \epsilon_t) = 0$$

$$Var(y_t) = Var(\epsilon_1) + Var(\epsilon_2) + \cdots + Var(\epsilon_t) = t\sigma^2$$

$$Cov(y_t, y_s) = \min(t, s)\sigma^2$$

* 내용 추가 : Time-series

시계열상관 검정

시계열의 자기상관(=계열상관) 여부를 파악하는 방법

→ 이는 잔차를 시각화하거나 통계적인 검정을 통해 확인할 수 있다.

- 더빈-왓슨 통계량(Durbin-Watson)

* 내용 추가 : Time-series

Unit Root Test(단위근 검정)

단위근 검정 (Unit Root Test)

- → 확률변수가 안정적인지? 불안정적인지? 확인하는 검정법 : 안정성 여부 검정하는 방법으로 공적분 검정에 앞서 하는 pre- Test
- 단위근 검정은 (비정상) 시계열에 대해 **확률적 추세 여부를 검정하며**(비정상 시계열인지 검정하는 문제로 귀결),
- **확률적 추세가 존재할 경우** 일반적으로 차분을 수행하여 정상성을 만족하도록 한다.
- **(비정상 시계열 특징)**
 - - 시점 t에 따라 평균이 다르거나,
 - - 시점 t에 따라 분산이 다르거나,
 - - 시차 h마다 공분산이 다르다.
 - - 추세나 계절성을 가진다.
-
- 추세는 **결정적 추세(deterministic trend)**(또는 비확률적 추세)와 **확률적 추세(stochastic trend)**로 나뉜다.

autoregressive(AR)(1) 하에서의 방법

$$y_t = \rho y_{t-1} + \nu_t, \nu_t \sim N(0, \sigma_\nu^2)$$

- $\rho = 1$ 이면 단위근을 가진다 → 비정상(non-stationary)
- $-1 \leq \rho \leq 1$: 정상(stationary) 시계열

시계열이 비정상인가 여부가설 검정

$$H_0 : \rho = 1, H_1 : |\rho| < 1$$

* 내용 추가 : Time-series

Unit Root Test(단위근 검정)

• DF(Dickey-Fuller Test) 검정 (τ 검정)

- DF 검정은 단위근 검정방법의 가장 근간이 되는 검정법
- 주어진 시계열 y_t 의 비정상성 여부는 주어진 시계열 y_t 를 차분하여 그를 다시 y_{t-1} 에 회귀하여 얻는 계수의 추정치가 0 인지 0 보다 작은지를 검정하는 문제로 귀결된다.
- → 즉, 1차분한 시계열이 정상시계열인지 비정상시계열인지 검정한다.
- → DF 검정은 임의보행 과정이 상수항(drift)를 가질 경우, 그리고 비확률 추세를 포함할 경우 등을 고려하여 귀무가설 검정
- (가정 : 오차항이 계열상관 되어 있지 않다.)

ADF(Augmented Dickey-Fuller Test) 검정

- ADF검정은 DF검정을 보완한 방법
- DF 검정을 위한 세가지 모형 설정 모두 오차항이 계열상관 되어 있지 않다는 가정이 전제된다.
- 오차항의 계열상관 되어 있을 경우, 이를 고려하기 위해 고안된 다음과 같은 모형 설정으로부터 오차항의 자기상관에 대한 문제를 제거한 단위근 검정을 ADF 검정이다.

KPSS(Kwiatkowski-Phillips-Schmidt-Shin Test) 검정

- 1종 오류의 발생가능성을 제거한 단위근 검정 방법
- DF 검정, ADF 검정의 귀무가설은 단위근이 존재한다는 것이나, KPSS 검정의 귀무가설은 정상 과정 (stationary process) 으로 검정 결과의 해석 시 유의할 필요가 있다.
- → 단위근 검정과 정상성 검정을 모두 수행함으로써 정상 시계열, 단위근 시계열, 또 확실히 식별하기 어려운 시계열을 구분하였다.
- KPSS 검정은 단위근의 부재가 정상성 여부에 대한 근거가 되지 못하며 대립가설이 채택되면 그 시계열은 trend-stationarity(추세를 제거하면 정상성이 되는 시계열)을 가진다고 할 수 있다.
- → KPSS 검정은 단위근을 가지지 않고 Trend- stationary인 시계열은 비정상 시계열이라고 판단 가능

Unit Root Test를 기반으로 한 장기 시계열 데이터의 non-stationary 발생에 따른 추세 변화 검정 및 시각화 연구1) 유재성

*, 주재걸** 고려대학교 컴퓨터학과 2019 참조

Time-series analysis

: 시계열(시간의 흐름에 따라 기록된 것) 자료(data)를 분석하고 여러 변수들간의 인과관계를 분석하는 방법론
→ 시계열분석과 횡단면분석의 성격을 결합하면 패널분석

univariate time series (단변량, 일변량 시계열 분석)

→ 시간대에 따라 변하는 추이를 자기의 과거데이터를 이용해서 분석

일변량(단변량) 정상시계열 모형 (univariate time series)

- 자기회귀(auto regression; AR)
- 이동평균법(moving average model :MA),
- naive methods, simple exponential smoothing, Box-Jenkins methods 등
스펙트럼 분석, 조건부이분산성(ARCH, GARCH) 모형 등
- 자기회귀 이동평균모형(autoregressive moving average model; 이하 ARMA)

stationary time series

약정상시계열(weak stationary time series) 세가지 조건 만족하는 시계열

- 1) 임의의 t 에 대하여 $E(X_t) = \mu$
- 2) 임의의 t 에 대하여 $\text{Var}(X_t) < \infty$
- 3) 임의의 t, h 에 대하여 $\text{Cov}(X_{t+h}, X_t) = \gamma(h)$ (즉, 공분산이 t 에 의존하지 않고 h 에만 의존한다.) $\gamma(h)$: 자기공분산함수(autocovariance function, ACVF)

1. 의의

ARIMA 모형은 비정상적(nonstationary) 시계열 자료에 대해 분석하는 방법

실제 ARMA 시계열 분석은 공분산 정상성(covariance stationary)을 만족시키는 과정을 거쳐 분석을 진행하게 되는데 이를 ARIMA 모형이라고 한다

. ARIMA 분석방법론은 시계열의 변동형태를 파악하고 이를 통해 예측이 가능하다는 장점으로 증권시장 등 경제분야에서 많이 응용되고 있다.

2. ARIMA모형 과정

모형 식별 (AR 1, AR2 , MA1, MA2) → ARIMA(1,1) → 모수 추정 → 검정 → 예측

3. 장점

1) 시계열 자료 외에 다른 자료가 없더라도 그 변동 상태를 확인할 수 있다

2) 어떤 시계열에도 적용이 가능하며 특히 시간의 흐름에 따라 자료의 변동이 빠를 때 민감하게 반영할 수 있다.

백색잡음 (White Noise)

→ 평균은 일정하고, 분산은 일정하며, 변수들간 공분산과 자기상관은 시점 t 에 의존하지 않고 각 변수들의 시점의 차이인 "시차"에만 의존한다는 정상성 조건 만족

: 변수들을 "백색잡음"으로 만드는 가장 효율적인 방법 → 차분(differencing)을 이용하는 방법 (=현재 변수에서 바로 전 차수의 변수를 차감하는 것)

일변량(단변량) 시계열모형에서, 외생 변수들의 영향을 받아 변화하는 경우가 많기 때문에 이를 고려하기 위해 개발된 모형

1. VAR (Vector AutoRegressive Model)

: 모형에는 전이함수모형에서 AR(1)만 고려하는 모형

- 일변량 자기회귀모형을 다변량 자기회귀모형으로 확장시킨 모형으로 예측 및 내생변수의 변화에 따른 효과 분석 등과 관련하여 활용
- 예측 + 특정 변수의 일시적 충격에 대한 효과를 모델링하기 위해 연립방정식 체계로 구성된 VAR 모형을 이용할 수 있음

Sims(1980) VAR 모형 기본 분석 모형의 한계를 보완하고자 고안한 모형

- 회귀분석: 시간 t 가 변하더라도 항상 일정하다는 가정이 있음
- 시계열 분석: 단변량 분석이기 때문에 변수 간의 상호작용을 고려하지 못함

(가정)

정상성(Stationarity) 가정

특정 t 시점에 관측된 값은 분포 값 가운데 단 하나의 값이므로, 특정 t 시점에 대한 분포를 추정하지 못한다는 한계를 지니기 때문 → 관측치 하나로 분포를 추정하는 것은 불가능하기 때문

- 정상성(Stationarity) 조건

VAR(p)

$$\begin{aligned} X_t &= C + \Theta_1 X_{t-1} + \cdots + \Theta_p X_{t-p} + \varepsilon_t \\ &= C + \sum_{i=1}^p \Theta_i X_{t-i} + \varepsilon_t \end{aligned}$$

- C 는 $(N \times 1)$ 상수벡터
- Θ_i 는 현시점의 변수와 시차변수들 간 시차회귀 계수인 $(N \times N)$ 의 행렬
- ε_t 는 $(N \times 1)$ 의 벡터 백색잡음 과정

VECM (Vector Error Correction Model)

VAR 모델의 경우 시계열 안정성을 위해 변수들을 차분하는 과정에서 변수들의 장기적인 관계에 대한 정보를 상실하여 모형 설정 오류가 발생

→ 벡터오차모형(VECM : Vector Error Correction Model)을 사용하여 이를 해결

공적분 검정: 단변량시계열과 달리 다변량시계열에서 단위근이 존재한다고 하여 바로 차분하는 것은 정보의 손실을 야기할 수 있으므로 장기 안정성이 존재 여부 테스트 → 요한슨 공적분 검정 많이 사용함

- 공적분이 존재하지 않음: 차분 변환하여 VAR모형으로 추정
 - 공적분이 존재함: VECM모형을 이용하여 추정
- 공적분이 존재할 때 이를 무시하고 VAR모형을 이용하면 추정결과가 VECM모형보다 좋지 않음

시차결정: 시차결정은 AIC, BIC, SBC 등 정보규준을 이용하여 결정

단변량시계열에서는 AR,MA항이 있는데 다변량에서는 추정계수를 줄여서 효율적인 추정을 하기 위하여 가역성원리를 이용하여 AR항만을 추정하거나, MA항을 추정할 수도 있다.

https://www.statsmodels.org/stable/vector_ar.html 파이썬 참조

공적분 (Cointegration)

1. 의미

두 시계열 사이에 존재하는 통계적 특성 중 하나로 단기적으로는 관계가 없어 보이나 장기적으로 일정 균형 관계

2. 공적분 관계 성립 조건

두 개 이상의 시계열(시간이 지남에 따라 변화하는 변수의 시간별 값을 기록한 자료)이 있을 때 다음과 같은 성질이 성립해야 한다.

- 1) 각각의 시계열들이 모두 동일한 order of integration*을 가진다.
- 2) 시계열들의 선형 결합으로 만들어진 새로운 시계열은 기존의 시계열들보다 더 낮은 order of integration을 가진다.

*order of integration : 어떤 시계열이 정상적(stationary)이 되기 위해 필요한 차분(difference) 횟수

10 Bayesian ML: Dynamic Sharpe Ratios and Pairs Trading

- Bayesian statistics allows us to quantify uncertainty about future events and refine estimates in a principled way as new information arrives.
- This dynamic approach adapts well to the evolving nature of financial markets.
- Bayesian approaches to ML enable new insights into the uncertainty around statistical metrics, parameter estimates, and predictions.
- The applications range from more granular risk management to dynamic updates of predictive models that incorporate changes in the market environment.

More specifically, this [chapter](#) covers:

- How Bayesian statistics applies to machine learning
- Probabilistic programming with PyMC3
- Defining and training machine learning models using PyMC3
- How to run state-of-the-art sampling methods to conduct approximate inference
- Bayesian ML applications to compute dynamic Sharpe ratios, dynamic pairs trading hedge ratios, and estimate stochastic volatility

베이즈 통계학(Bayesian statistics) **베이즈 추론 (Bayesian inference)

베이즈 통계학(Bayesian statistics)

- 하나의 사건에서의 믿음의 정도 (degree of belief)를 확률로 나타내는 베이즈 확률론에 기반한 통계학 이론
- 믿음의 정도는 이전 실험에 대한 결과, 또는 그 사건에 대한 개인적 믿음 등, 그 사건에 대한 사전 지식에 기반할 수 있다.
- 기존에 가지고 있었던 어떠한 '선입견'의 개념을 수치화해서 계산에 넣을 수 있다.→ 이러한 사고방식을 바로 베이저안 통계학→ 다양한 머신러닝 알고리즘의 근간
- 통계적 머신러닝이란 결국 이 베이저안 통계학의 원칙 하에서, 선입견을 조금씩 수정하는 과정

베이즈 추론

- 통계적 추론의 한 방법으로, 추론 대상의 사전 확률과 추가적인 정보를 통해 해당 대상의 사후 확률을 추론하는 방법
- 베이즈 추론은 베이즈 확률론을 기반으로 하며, 이는 추론하는 대상을 확률변수로 보아 그 변수의 확률분포를 추정하는 것을 의미

베이즈 정리(Bayes' theorem)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

B라는 사건이 일어난 상태에서 A가 발생할 확률을 계산하는 방법

→ 여기서 베이저안 통계적 추론을 하는데, 통계적 추론은 빈도 확률론자(frequentist)의 관점을 사용한다.

이는 시행의 결과를 보고, 가장 그럴듯한 확률을 추정하는 방법이다.

→ 그러나 현실은 선입견을 갖고 추론(동전 앞면 나올 확률 1/2) 하려 한다. 이러한 '선입견'을 수치화 해서 계산에 포함시키고자 하는게 베이저안 확률론자의 관점이 가지고 있는 기본 철학

→ 실제 시행 결과 가능도(likelihood) 가장 높게 나올 경우를 추정하는 최대 가능도 추정 (maximum likelihood estimation: MLE)와 선입견을 반영해서(0.5) 추정을 하는 과정을 최대 사후 확률 추정(maximum a posteriori)

→ → 시행을 여러번 하면서 차츰 선입견을 업데이트 하는 과정을 베이저안 갱신(Bayesian update)→ 통계적 머신러닝에서의 "학습"

PyMC (이전의 PyMC3)

고급 Markov 체인 Monte Carlo 및 Variational Fitting 알고리즘에 중점을 둔 **베이지안 통계 모델링** 및 **확률적 기계학습**을 위한 [Python](#) 패키지

<https://docs.pymc.io/en/v3/>

**주의: python3.6 보다 높은 버전에서는 에러 발생 가능성

Installation

- [Instructions for Linux](#)
- [Instructions for MacOS](#)
- [Instructions for Windows](#)

PyMC implements non-gradient-based and gradient-based [Markov chain Monte Carlo](#) (MCMC) algorithms for Bayesian inference and stochastic, gradient-based [variational Bayesian methods](#) for approximate Bayesian inference.

•MCMC-based algorithms:

- No-U-Turn sampler^[28] (NUTS), a variant of [Hamiltonian Monte Carlo](#) and PyMC's default engine for continuous variables
- [Metropolis–Hastings](#), PyMC's default engine for discrete variables
- Sequential Monte Carlo for static posteriors
- Sequential Monte Carlo for [Approximate Bayesian computation](#)

•Variational inference algorithms:

- Black-box Variational Inference

11 Random Forests: A Long-Short Strategy for Japanese Stocks

In short, this chapter covers:

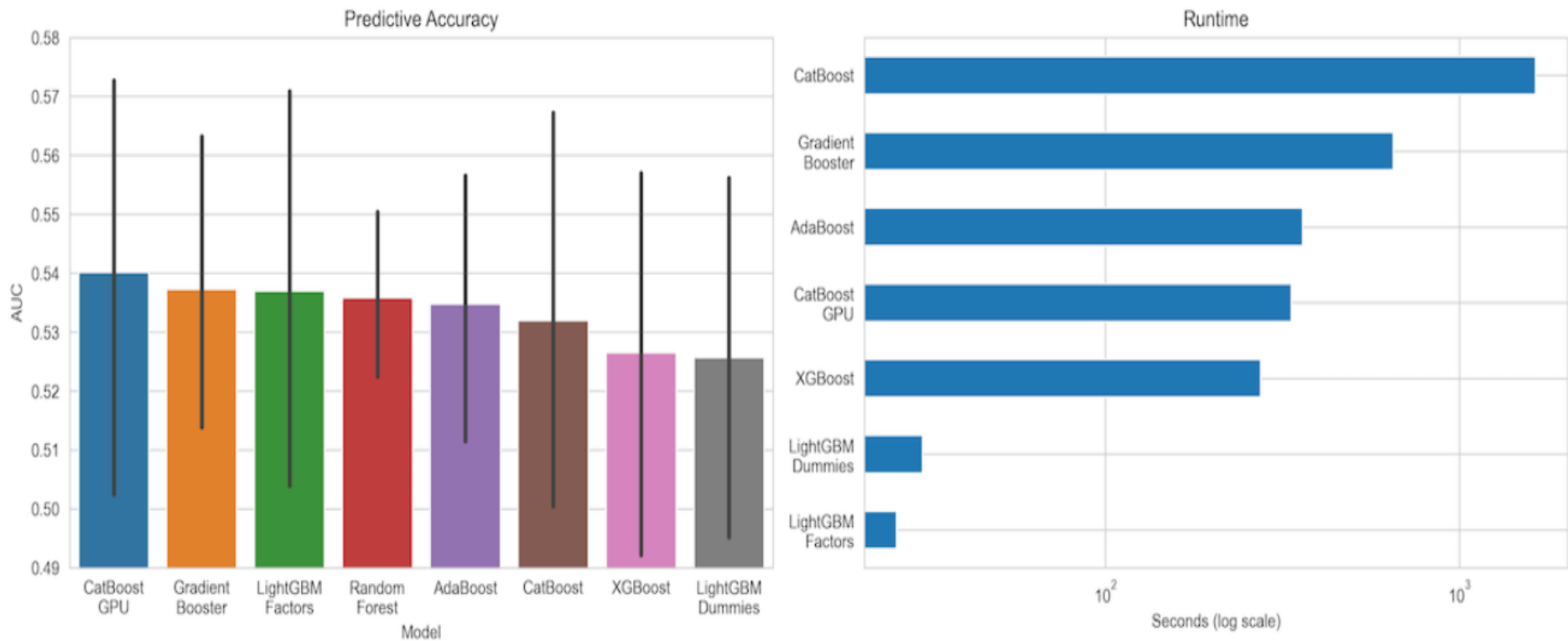
- Use decision trees for regression and classification
- Gain insights from decision trees and visualize the rules learned from the data
- Understand why ensemble models tend to deliver superior results
- Use bootstrap aggregation to address the overfitting challenges of decision trees
- Train, tune, and interpret random forests
- Employ a random forest to design and evaluate a profitable trading strategy

12 Boosting your Trading Strategy

Gradient boosting is an alternative tree-based ensemble algorithm that often produces better results than random forests.

The critical difference is that boosting modifies the data used to train each tree based on the cumulative errors made by the model.

While random forests train many trees independently using random subsets of the data, boosting proceeds sequentially and reweights the data.



12 Boosting your Trading Strategy

More specifically, we will cover the following topics:

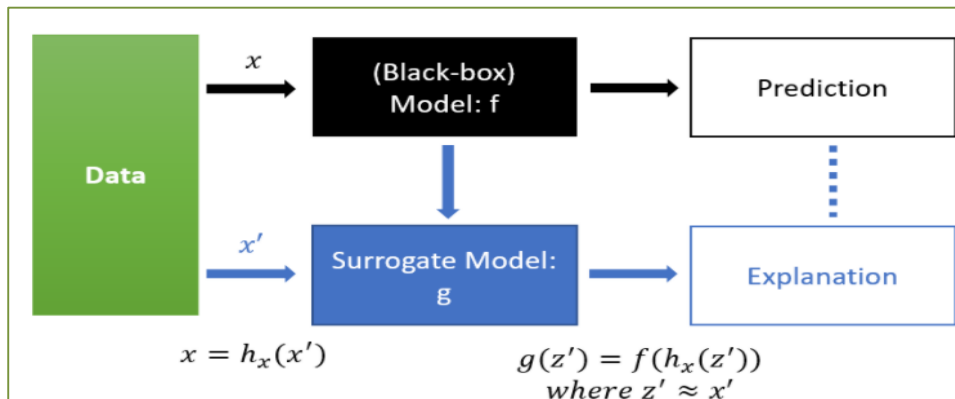
- How does **boosting** differ from **bagging**, and how did **gradient boosting** evolve from **adaptive boosting**,
- Design and tune **adaptive and gradient boosting models** with **scikit-learn**,
- Build, optimize, and evaluate **gradient boosting models** on large datasets with the state-of-the-art implementations **XGBoost, LightGBM, and CatBoost**,
- Interpreting and gaining insights from **gradient boosting models** using [SHAP](#) values, and
- Using boosting with high-frequency data to design an **intraday strategy**.

** SHAP(SHapley Additive exPlanation)

모델링을 하면서 반드시 그 원인 인자를 찾고, 얼마나 결과에 영향을 주었는지를 파악이 필요한 경우

→ 결과를 "설명"하는 것이 중요해지면서 XAI(eXplainable AI)가 관심이 커지면서 SHAP은 Shapley Value를 근간으로 하는 XAI 이다.

→ SHAP은 Local Explanation을 기반으로 하여, 데이터의 전체적인 영역에 대한 해석(Global Surrogate)이 가능



SHAP Values

A Unified Approach to Interpreting Model Predictions(2017) 논문

SHAP values를 특성 중요도의 "통합된 측정 방식(unified measure)" 제안

→ Shapely Value의 조건부평균(Conditional Expectation)으로, simplified input을 정의하기 위해 original model인 f 의 값이 아닌 f 의 조건부평균을 계산

→ SHAP values는 어떤 특성의 조건부 조건에서 해당 특성이 모델 예측치의 변화를 가져오는 정도

→ SHAP Values는 Feature Attribution의 **3가지 특징(Local Accuracy, Missingness, Consistency)**을 만족하는 유니크한 가산적 특성 중요도 측정(additive feature importance measure) 방식을 제공

13 Data-Driven Risk Factors and Asset Allocation with Unsupervised Learning

Dimensionality reduction and clustering are the main tasks for unsupervised learning:

- Dimensionality reduction transforms the existing features into a new, smaller set while minimizing the loss of information.
- A broad range of algorithms exists that differ by how they measure the loss of information, whether they apply linear or non-linear transformations or the constraints they impose on the new feature set.
- Clustering algorithms identify and group similar observations or features instead of identifying new features.

More specifically, this [chapter](#) covers:

- How principal and independent component analysis (PCA and ICA) perform linear dimensionality reduction
- Identifying data-driven risk factors and eigenportfolios from asset returns using PCA
- Effectively visualizing nonlinear, high-dimensional data using manifold learning
- Using T-SNE and UMAP to explore high-dimensional image data
- How k-means, hierarchical, and density-based clustering algorithms work
- Using agglomerative clustering to build robust portfolios with hierarchical risk parity

* t-SNE(t-distributed stochastic neighbor embedding)

t-SNE

- t 분포를 이용하여 멀리 떨어진 포인트와의 거리를 보존하는 것보다 고차원의 원공간에 존재하는 data x의 이웃 (neighbor)간의 distance를 최대한 집중 보존하며 대응되는 저차원의 y를 학습하는 비지도학습 방법론
- dimensionality reduction과 visualization에 많이 쓰이는 고차원 data 시각화 하는데 많이 유용한 알고리즘

Stochastic neighbor embedding (SNE)

거리정보를 확률적으로 나타내기 때문에 붙임.

$$p_{j|i} = \frac{e^{-\frac{|x_i - x_j|^2}{2\sigma_i^2}}}{\sum_k e^{-\frac{|x_i - x_k|^2}{2\sigma_i^2}}}$$
$$q_{j|i} = \frac{e^{-|y_i - y_j|^2}}{\sum_k e^{-|y_i - y_k|^2}}$$

두 확률 분포를 같게 만들기 위해서 두 확률 분포의 유사도를 측정하는 분포인 Kullback-Leibler Divergence를 cost function으로 사용해서 이를 최소화 하는 방향으로 학습을 진행.

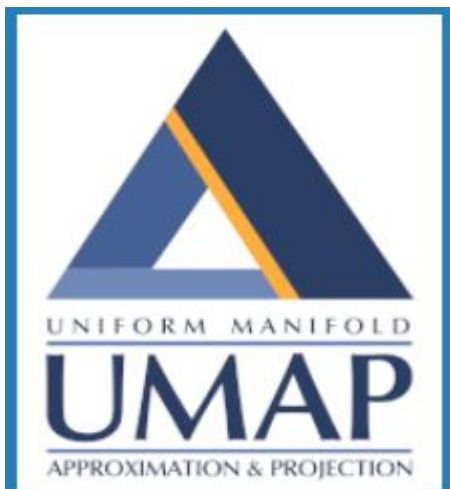
(단점) 시간이 오래 걸림, 2, 3차원으로만 축소 가능, Sparse한 matrices에 바로 적용 불가(선행적으로 PCA와 SVD 이루어져야 함)

KL Divergence식과 gradient

$$\sum_i KL(P_i || Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

UMAP (Uniform Manifold Approximation and Projection)

- t-SNE보다 더욱 빠르면서도 데이터 공간을 잘 분리하는 UMAP(Uniform Manifold Approximation and Projection)은 **비선형 차원 축소**를 위해 제안
- 매우 큰 데이터셋을 빠르게 처리할 수 있고, 희소 행렬 데이터에 적합
- t-SNE에 비해 다른 머신러닝 모델에서 새로운 데이터가 들어왔을 때 즉시 embedding 가능 장점
- 비지도학습



https://umap-learn.readthedocs.io/en/latest/how_umap_works.html 참고

14 Text Data for Trading: Sentiment Analysis

Text data is very rich in content but highly unstructured so that it requires more preprocessing to enable an ML algorithm to extract relevant information.

A key challenge consists of converting text into a numerical format without losing its meaning.

This [chapter](#) shows how to represent documents as vectors of token counts by creating a document-term matrix that, in turn, serves as input for text classification and sentiment analysis.

It also introduces the Naïve Bayes algorithm and compares its performance to linear and tree-based models.

In particular, in this chapter covers:

- What the fundamental NLP workflow looks like
- How to build a multilingual feature extraction pipeline using spaCy and TextBlob
- Performing NLP tasks like part-of-speech tagging or named entity recognition
- Converting tokens to numbers using the document-term matrix
- Classifying news using the naive Bayes model
- How to perform sentiment analysis using different ML algorithms

자연어 처리(NLP) 텍스트분석을 위한 파이썬 라이브러리

NLTK(Natural Language Toolkit)

가장 널리 알려진 고성능 파이썬 NLP 라이브러리

[많은 코포라\(Corpora, 데이터 세트\)와 훈련된 모델](#)을 NLTK와 함께 사용해 즉시 NLTK에 대한 실험을 시작할 수 있게 해준다.

문서에 설명되어 있듯, NLTK는 텍스트를 다루기 위한 다양한 도구들을 제공한다. 분류, 토큰화, 스템밍(Stemming), 태깅, 파싱, 시멘틱 추론을 예로 들 수 있다.

스탠포드 태거(Stanford Tagger), TADM, MEGAM 같은 [써드파티 도구들 가운데 일부](#)도 지원

Gensim

통계적 의미론(statistical semantics)에 초점이 맞춰져 있는데, 문서의 구조를 분석한 후 유사성을 기준으로 다른 문서에 점수를 부여한다.

Gensim은 문서를 분석 엔진으로 스트리밍하고, 점진적으로 [비지도 학습](#)을 수행해 아주 큰 텍스트 본문을 처리할 수 있게 해준다. 또 각기 다른 시나리오에 부합하는 여러 종류의 모델을 생성할 수 있다.

Word2Vec, Doc2Vec, FastText 및 Latent Dirichlet Allocation

CoreNLP

스탠포드 대학(Stanford University)이 만든 [CoreNLP 라이브러리](#)는 NLP 예측 및 분석 작업을 대규모로 수행할 수 있게 해주는 실용 단계의 NLP 솔루션이다. CoreNLP는 자바(Java)로 작성됐지만, 이를 위한 API와 여러 파이썬 패키지가 등장해 있는 상태다. [Stanza](#)로 불리는 네이티브 NLP 라이브러리가 그 중 하나다.

문법 태깅, 명명 엔티티 인식, 파싱, 구문 분석, 감성 분석 등 [많은 언어 관련 도구](#)들이 CoreNLP에 포함

Pattern

인기 웹사이트를 스크레이핑(Scrape)해서 분석하는 경우 많이 이용.

Pattern에는 구글, 위키피디아, 트위터, 페이스북, 제네릭 RSS 등 인기 웹 서비스와 소소를 스크레이핑 할 수 있는 도구가 탑재되어 있다. 모두 파이썬 모듈로 이용할 수 있다. 이런 사이트 각각에서 데이터를 가져오기 위해 무언가를 처음부터 만들 필요가 없다. 이후 감성 분석 같은 많이 사용되는 다양한 NLP 작업을 데이터에 수행할 수 있다.

Pattern은 직접 NLP 함수, n-gram 검색, 벡터, 그래프를 사용할 수 있는 기능도 지원

자연어 처리(NLP) 텍스트분석을 위한 파이썬 라이브러리

<https://pypi.org/project/spacytextblob/>

SpaCy

편의성에는 파이썬, 속도에는 Cython를 활용하는 [SpaCy](#)는 '산업에 강한 NLP'

속도와 모델 크기, 정확성 측면에서 NLTK, CoreNLP 등의 다른 경쟁자를 앞선다고 [주장한다](#).

SpaCy에는 경쟁 프레임워크에서 제공되는 거의 대부분 기능이 포함되어 있다.

음성 태깅, 종속성 파싱, 명명 엔티티 인식, 토큰화, 감성 세그멘테이션, 규칙 기반 매칭 작업, 워드 벡터 등을 예로 들 수 있다.

SpaCy에는 GPU 작업 최적화 기능도 지원한다. 연산을 가속화하고, 복제를 막기 위해 GPU에 데이터를 저장한다.

spacytextblob 3.0.1

설정 마법사가 윈도우와 리눅스, 맥OS, 기타 여러 파이썬 환경(pip와 conda 등)을 위한 명령줄 설치 작업을 생성한다. 언어 모델은 파이썬 패키지로 설치된다. 따라서 애플리케이션의 종속성 리스트의 일부로 추적할 수 있다.

TextBlob

[TextBlob](#)은 Pattern과 NLTK 라이브러리의 친화적인 프론트엔드이다. 두 라이브러리를 고수준의 사용하기 쉬운 인터페이스로 포장한다. TextBlob을 사용하면 패턴과 NLTK의 복잡함에 어려움을 겪는 시간이 줄어들고, 결과를 얻는 시간이 늘어난다.

TextBlob은 네이티브 파이썬 객체와 구문을 활용해 원활한 작업을 돕는다. [퀵스타트 예제](#)는 처리할 텍스트를 문자열로 단순히 처리하는 방법을 보여주며, 음성 일부 태깅 등 많이 사용하는 NLP 방법을 문자열 객체에서 사용할 수 있다.

Textblob의 또 다른 장점은 더 자신감을 얻었을 때 기능성을 변경할 수 있다는 것이다. 감성 분석 시스템이나 토큰화 같은 많은 기본 구성요소들을 필요에 따라 [교체할 수 있다](#). 또한 감성 분석 도구나 분류자 등 구성요소를 결합하는 고수준 객체를 생성하고, 최소한의 노력으로 이를 재사용할 수 있다.

15 Topic Modeling: Summarizing Financial News

This [chapter](#) uses unsupervised learning to model latent topics and extract hidden themes from documents.

These themes can generate detailed insights into a large corpus of financial reports.

Topic models automate the creation of sophisticated, interpretable text features that, in turn, can help extract trading signals from extensive collections of texts.

They speed up document review, enable the clustering of similar documents, and produce annotations useful for predictive modeling.

Applications include identifying critical themes in company disclosures, earnings call transcripts or contracts, and annotation based on sentiment analysis or using returns of related assets.

More specifically, it covers:

- How topic modeling has evolved, what it achieves, and why it matters
- Reducing the dimensionality of the DTM using latent semantic indexing
- Extracting topics with probabilistic latent semantic analysis (pLSA)
- How latent Dirichlet allocation (LDA) improves pLSA to become the most popular topic model
- Visualizing and evaluating topic modeling results -
- Running LDA using scikit-learn and gensim
- How to apply topic modeling to collections of earnings calls and financial news articles

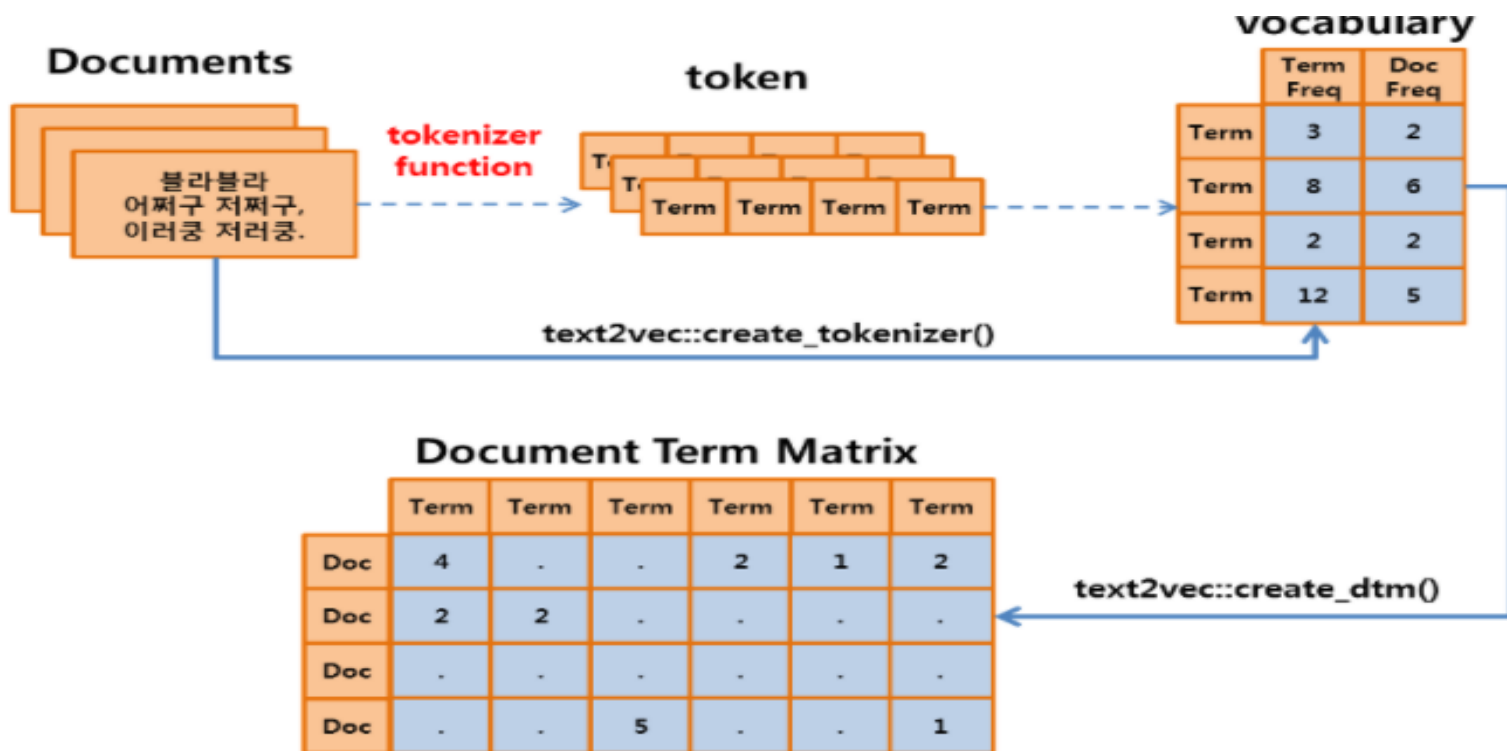
Reducing the dimensionality of the DTM

문서단어행렬 (DTM: Document Term Matrix)

모델링을 위해서 비정형데이터인 문서(documents) 텍스트 데이터를 벡터(vector)로 변환해야 한다.

→ 연산 속도의 개선을 위해서 벡터화(vectorization) 구조로 연산을 해야 하기 때문에 벡터화(Vectorization) 연산을 수행하기 위한 구조로 변환이 필요

문서단어행렬(DTM)생성 과정



Reducing the dimensionality of the DTM

차원 축소 (Dimensionality Reduction)

1. 차원 축소 목적

1) 시간 복잡도 (time complexity)와 공간 복잡도 (space complexity)가 감소

--> 효율적

2) 과적합 문제 차단

다차원의 데이터로 학습시킨 머신러닝 모델은 내부의 파라미터도 매우 복잡하게 형성되기 때문에 과적합 문제 발생

--> 입력 데이터의 차원을 줄여서 학습을 시키면 모델이 비교적 간단해지고, 그러면 적은 데이터 셋에 대해 안정적(robust)인 결과

3) 간결한 모델일수록 이해하기에 편하다. (interpretable)

모델이 내놓은 결과를 2차원이나 3차원의 그림으로 축소해 드러내어서, 이해와 정보 전달이 용이

2. 차원 축소 방식

정보 손실 문제 방지 해소를 고려

1) feature selection 방법

: 데이터 열(column)을 하나씩 골라내는 것, Forward, Backward

2) feature extraction 방법

여러 무더기의 데이터 열(column)을 압축하 방법을 제안

Reducing the dimensionality of the DTM

차원축소 (Dimensionality Reduction) 사례

(data)

1) feature selection 방법

- -총 5개의 열이 입력 데이터(각각 x_0 부터 x_4 까지) $d = 5$, -결과치(정답 레이블) 마지막 열(answer열)

◆ (Forward search 방법)

1) F를 비어있는 공집합으로 둔다. : 만들어진 결과 테이블을 F라고 명명

2) E(error)를 구하는데 x_j (=j 번째 열) 을 F에다가 (임시로) 넣은 후에(합집합), 머신러닝 모델을 돌려서 E(error)를 확인.

이때 x_0 부터 x_4 까지의 모든 열에 대해서 반복해서 머신러닝 모델 실행, → 그중에 가장 성능이 좋은 (= E(error)를 가장 작은) x_j 를 발견

3) 2)에서 찾은 결과를 결과 테이블인 F의 결과로 (영구적으로) 등록

4) 2)와 3)의 계산을 반복해서 실행--> 최종적으로 k개의 열로 이루어진 테이블

첫 번째 반복에서는 d회의 머신러닝 훈련(training), 두 번째 반복에서는 (영구적으로 뽑힌 열을 제외하고) d-1회 의 훈련을 하고, k번째 반복에서는 d-(k-1)회의 훈련 실행

◆ (Backword search 방법) - Forward search 방법을 반대로 실행

1) F를 원래 데이터 전체로 놓다.

2) E(error)를 구하는데, F로부터 열을 하나씩 빼면서 ML모델을 돌려 E(error)를 확인하고, 퇴출시키면 모델의 성능이 가장 좋게 나오는 x_j 열을 알아낼 수 있다.

3) 2)에서 찾은 결과를 F로부터 퇴출

4) k개의 열로 이루어진 테이블을 원하므로--> 첫 번째 퇴출에서는 전체 데이터 열 개수인 d회를 계산, ..., k번째 반복에서는 k+1 회의 계산이 이루어진다.

Reducing the dimensionality of the DTM

차원 축소 (Dimensionality Reduction)

2) feature extraction 방법

여러 무더기의 데이터 열(column)을 압축하는 방법을 제안

정답 레이블(answer)을 사용하지 않는 unsupervised 방법과 정답 레이블(answer)을 사용하는 supervised 방법으로 구분

LDA : Latent Dirichlet Allocation

1. LDA란

문서의 집합으로부터 어떤 토픽이 존재하는지를 알아내기 위한 알고리즘

빈도수 기반의 표현 방법인 BoW의 행렬 DTM 또는 TF-IDF 행렬을 입력으로 하는데, 이로부터 알 수 있는 사실은 LDA는 단어의 순서는 고려하지 않는 다는 것.

2. 준비

- 1) 문서에 사용할 단어의 개수 N 사전 결정
- 2) 문서에 사용할 토픽의 혼합을 확률 분포에 기반하여 결정.
- 3) 문서에 사용할 각 단어를 결정
 - 토픽 분포에서 토픽 T 를 확률적으로 선택
 - 선택한 토픽 T 에서 단어의 출현 확률 분포에 기반해 문서에 사용할 단어 선택
- 4) 3)을 반복 → 문서 완성

이러한 과정을 통해 문서가 작성되었다는 가정 하에 LDA는 토픽을 뽑아내기 위하여 위 과정을 역으로 추적하는 역공학(reverse engineering)을 수행.

3. LDA의 수행하기

- 1) 사용자 → 알고리즘, 토픽의 개수 k 제공
- 2) 모든 단어를 k 개 중 하나의 토픽에 할당
- 3) 이제 모든 문서의 모든 단어에 대해서 아래 사항 반복 진행 (iterative)
 - $p(\text{topic } t \mid \text{document } d)$: 문서 d 의 단어들 중 토픽 t 에 해당하는 단어들의 비율
 - $p(\text{word } w \mid \text{topic } t)$: 토픽 t 중 단어 w 를 갖고 있는 모든 문서들 비율

16 Word embeddings for Earnings Calls and SEC Filings

This [chapter](#) uses neural networks to learn a vector representation of individual semantic units like a word or a paragraph. These vectors are dense with a few hundred real-valued entries, compared to the higher-dimensional sparse vectors of the bag-of-words model. As a result, these vectors embed or locate each semantic unit in a continuous vector space.

Embeddings result from training a model to relate tokens to their context with the benefit that similar usage implies a similar vector. As a result, they encode semantic aspects like relationships among words through their relative location.

They are powerful features that we will use with deep learning models in the following chapters.

More specifically, in this chapter, we will cover:

- What word embeddings are and how they capture semantic information
- How to obtain and use pre-trained word vectors
- Which network architectures are most effective at training word2vec models
- How to train a [word2vec model](#) using [TensorFlow and gensim](#)
- Visualizing and evaluating the quality of word vectors
- How to train a word2vec model on SEC filings to predict stock price moves
- How [doc2vec](#) extends word2vec and helps with sentiment analysis
- Why the transformer's attention mechanism had such an impact on NLP
- How to fine-tune pre-trained [BERT](#) models on financial data

17 Deep Learning for Trading

This [chapter](#) presents feedforward neural networks (NN) and demonstrates how to efficiently train large models using backpropagation while managing the risks of overfitting.

It also shows how to use [TensorFlow 2.0](#) and [PyTorch](#) and how to optimize a NN architecture to generate trading signals. In the following chapters, we will build on this foundation to apply various architectures to different investment applications with a focus on alternative data.

These include recurrent NN tailored to sequential data like time series or natural language and convolutional NN, particularly well suited to image data.

We will also cover deep unsupervised learning, such as how to create synthetic data using Generative Adversarial Networks ([GAN](#)).

Moreover, we will discuss reinforcement learning to train agents that interactively learn from their environment.

In particular, this chapter will cover

- How DL solves AI challenges in complex domains
- Key innovations that have propelled DL to its current popularity
- How feedforward networks learn representations from data
- Designing and training deep neural networks (NNs) in Python
- Implementing deep NNs using [Keras](#), TensorFlow, and PyTorch
- Building and tuning a deep NN to predict asset returns
- Designing and backtesting a trading strategy based on deep NN signals

18 CNN for Financial Time Series and Satellite Images

CNN architectures continue to evolve.

This chapter describes building blocks common to successful applications, demonstrates how transfer learning can speed up learning, and how to use CNNs for object detection.

CNNs can generate trading signals from images or time-series data.

Satellite data can anticipate commodity trends via aerial images of agricultural areas, mines, or transport networks. Camera footage can help predict consumer activity; we show how to build a CNN that classifies economic activity in satellite images.

CNNs can also deliver high-quality time-series classification results by exploiting their structural similarity with images, and we design a strategy based on time-series data formatted like images.

More specifically, this [chapter](#) covers:

- How CNNs employ several building blocks to efficiently model grid-like data
- Training, tuning and regularizing CNNs for images and time series data using TensorFlow
- Using transfer learning to streamline CNNs, even with fewer data
- Designing a trading strategy using return predictions by a CNN trained on time-series data formatted like images
- How to classify economic activity based on satellite images

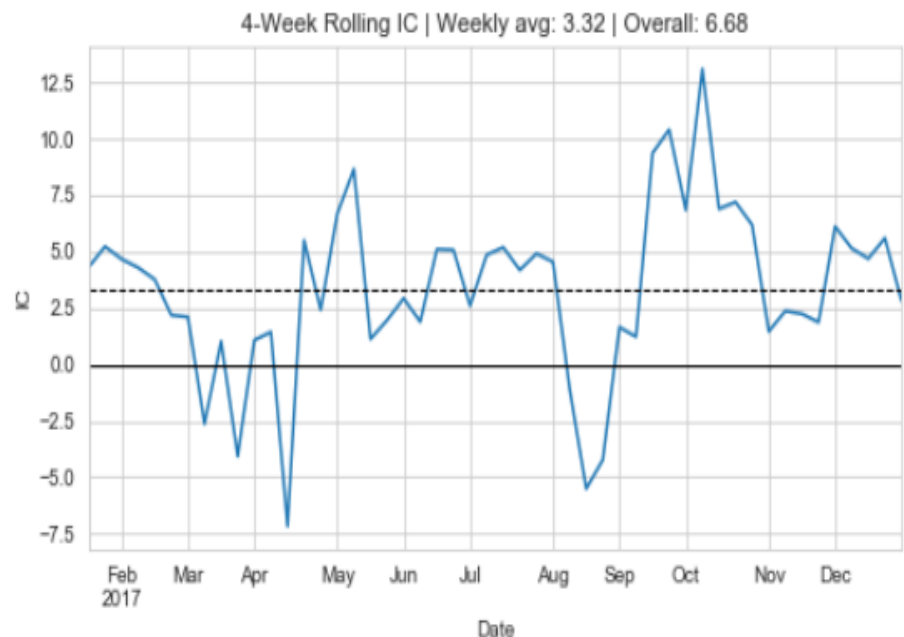
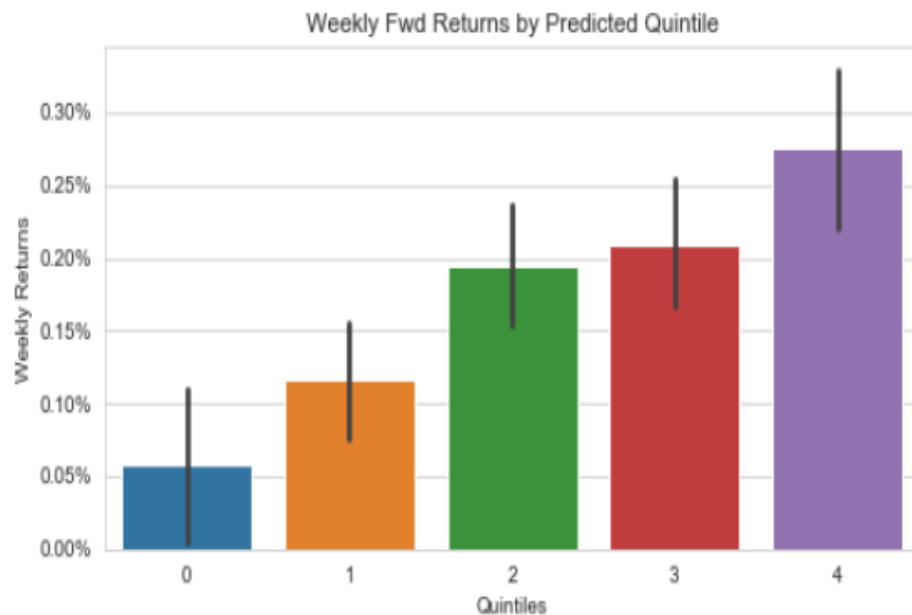
19 RNN for Multivariate Time Series and Sentiment Analysis

Recurrent neural networks (RNNs) compute each output as a function of the previous output and new data, effectively creating a model with memory that shares parameters across a deeper computational graph. Prominent architectures include Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) that address the challenges of learning long-range dependencies.

RNNs are designed to map one or more input sequences to one or more output sequences and are particularly well suited to natural language.

They can also be applied to univariate and multivariate time series to predict market or fundamental data.

This chapter covers how RNN can model alternative text data using the word embeddings that we covered in Chapter 16 to classify the sentiment expressed in documents.



19 RNN for Multivariate Time Series and Sentiment Analysis

More specifically, this chapter addresses:

- How recurrent connections allow RNNs to memorize patterns and model a hidden state
- Unrolling and analyzing the computational graph of RNNs
- How gated units learn to regulate RNN memory from data to enable long-range dependencies
- Designing and training RNNs for univariate and multivariate time series in Python
- How to learn word embeddings or use pretrained word vectors for sentiment analysis with RNNs
- Building a bidirectional RNN to predict stock returns using custom word embeddings

20 Autoencoders for Conditional Risk Factors and Asset Pricing

This [chapter](#) shows how to leverage unsupervised deep learning for trading.

We also discuss autoencoders, namely, a neural network trained to reproduce the input while learning a new representation encoded by the parameters of a hidden layer.

Autoencoders have long been used for nonlinear dimensionality reduction, leveraging the NN architectures we covered in the last three chapters.

We replicate a recent AQR paper that shows how autoencoders can underpin a trading strategy.

We will use a deep neural network that relies on an autoencoder to extract risk factors and predict equity returns, conditioned on a range of equity attributes.



More specifically, in this chapter you will learn about:

- Which types of autoencoders are of practical use and how they work
- Building and training autoencoders using Python
- Using autoencoders to extract data-driven risk factors that take into account asset characteristics to predict returns

21 Generative Adversarial Nets for Synthetic Time Series Data

This chapter introduces **generative adversarial networks (GAN)**. GANs train a generator and a discriminator network in a competitive setting so that the generator learns to produce samples that the discriminator cannot distinguish from a given class of training data. The goal is to yield a generative model capable of producing synthetic samples representative of this class. While most popular with image data, GANs have also been used to generate synthetic time-series data in the medical domain. Subsequent experiments with financial data explored whether GANs can produce alternative price trajectories useful for ML training or strategy backtests. We replicate the 2019 NeurIPS Time-Series GAN paper to illustrate the approach and demonstrate the results.

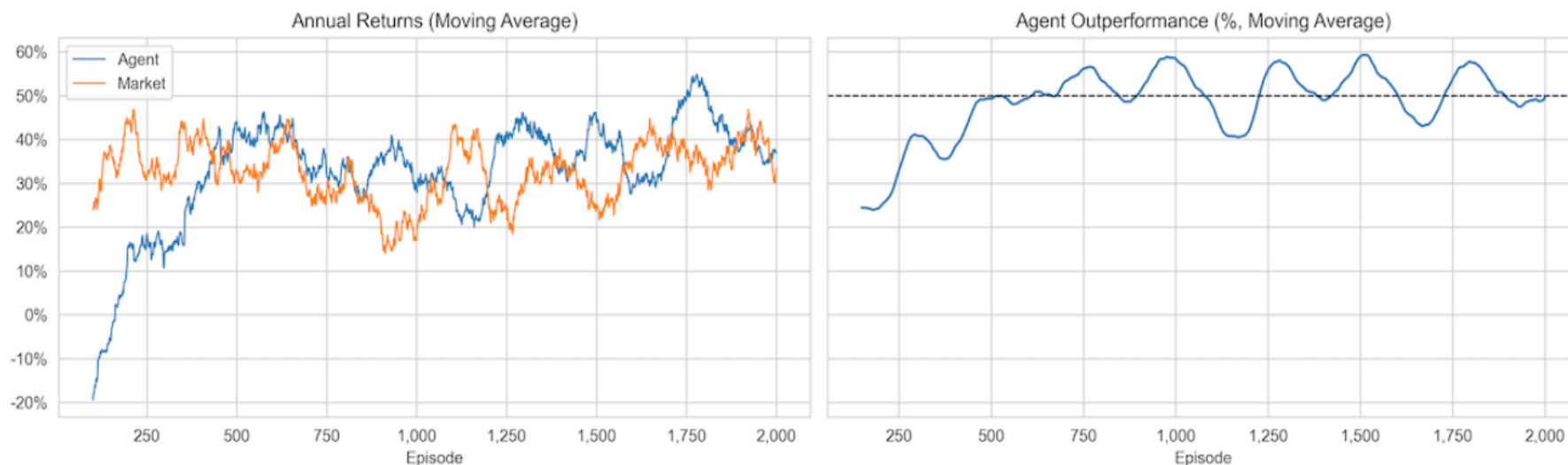
More specifically, in this chapter you will learn about:

- How GANs work, why they are useful, and how they could be applied to trading
- Designing and training GANs using TensorFlow 2
- Generating synthetic financial data to expand the inputs available for training ML models and backtesting

22 Deep Reinforcement Learning: Building a Trading Agent

Reinforcement Learning (RL) models goal-directed learning by an agent that interacts with a stochastic environment. RL optimizes the agent's decisions concerning a long-term objective by learning the value of states and actions from a reward signal. The ultimate goal is to derive a policy that encodes behavioral rules and maps states to actions.

This [chapter](#) shows how to formulate and solve an RL problem. It covers model-based and model-free methods, introduces the OpenAI Gym environment, and combines deep learning with RL to train an agent that navigates a complex environment. Finally, we'll show you how to adapt RL to algorithmic trading by modeling an agent that interacts with the financial market while trying to optimize an objective function.



22 Deep Reinforcement Learning: Building a Trading Agent

More specifically, this chapter will cover:

- Define a Markov decision problem (MDP)
- Use value and policy iteration to solve an MDP
- Apply Q-learning in an environment with discrete states and actions
- Build and train a deep Q-learning agent in a continuous environment
- Use the OpenAI Gym to design a custom market environment and train an RL agent to trade stocks

Machine Learning for Asset Managers (Elements in Quantitative Finance)

[Marcos M. López de Prado](#), *Cornell University,
New York*



<https://github.com/emoen/Machine-Learning-for-Asset-Managers>

코세라 무료강좌

<https://www.coursera.org/learn/python-machine-learning-for-investment-management>

p-값

- 1700년대 (Brian and Jaission 2007)로 거슬러 올라가는 개념인 p 값을 통해서다.
- p-값은 해당 변수와 관련된 실제 계수가 0일 때 우리가 추정했던 것과 같거나 더 극단적인 결과를 얻었을 확률을 계량화한다.
- 이는 데이터가 설정된 통계 모델과 얼마나 일치하지 않는지를 나타낸다.
- p-값은 귀무가설이나 대립가설이 참이 아니거나 데이터가 랜덤일 확률을 측정하지 않는다.
- p-값은 효과의 크기나 결과의 유의성을 측정하지 않는다.
- p-값의 오용은 매우 광범위하게 퍼져 있어서 미국통계협회는 통계적 유의성의 척도로서 앞으로 그들의 적용을 권장하고 있지 않다(Wasserstein et al. 2019).
- 금융에서의 수십 년간의 실증 연구에 의문을 제기한다.
- p-값의 대안을 찾으려면 우선 p-값의 함정을 이해해야 한다.

- **p-값 결함**

- 첫 번째 결함은 앞에서 설명한 강력한 가정에 의존한다는 것이다.

→ 그러한 가정들이 정확하지 않을 때 p-값 계수의 참 값이 0이더라도 p-값이 낮을 수 있고(거짓 양성), 계수의 참 값이 0이 아님에도 p-값이 높을 수 있다(거짓 음성).

- 두 번째 결함은 높은 다중 공선(상호 상관) 설명 변수에 대해 p-값을 강건하게 추정할 수 없다는 것이다.

→ 다중 공선 시스템에서 전통적인 회귀 분석 방법은 중복 설명 변수를 구별할 수 없으므로 관련 p-값 간의 대체 효과가 발생한다.

- 세 번째 결함은 완전히 관련이 없는 확률을 평가한다는 것이다.

- 네 번째 결함은 샘플의 유의성을 평가한다는 것이다.

- 전체 샘플은 계수 추정과 유의성 결정이라는 두 가지 과제를 해결하는 데 사용된다. 따라서 p-값은 샘플 외 설명(즉 예측) 값이 없는 변수에 대해 낮을 수 있다(즉 유의할 수 있다). 동일한 데이터셋에 대해 여러 번의 샘플 내 테스트를 실행하면 잘못된 발견이 발생할 가능성이 높으며, 이는 p - 해킹으로 알려진 관행이다.

Do not carry all the eggs (investment)
in one basket(**one algorithm**)

y.b. Jeon

I will remember the eyes of all of you who looked at me.



We can only see a short distance ahead, but we can see plenty there that needs to be don

Alan Mathison Turing