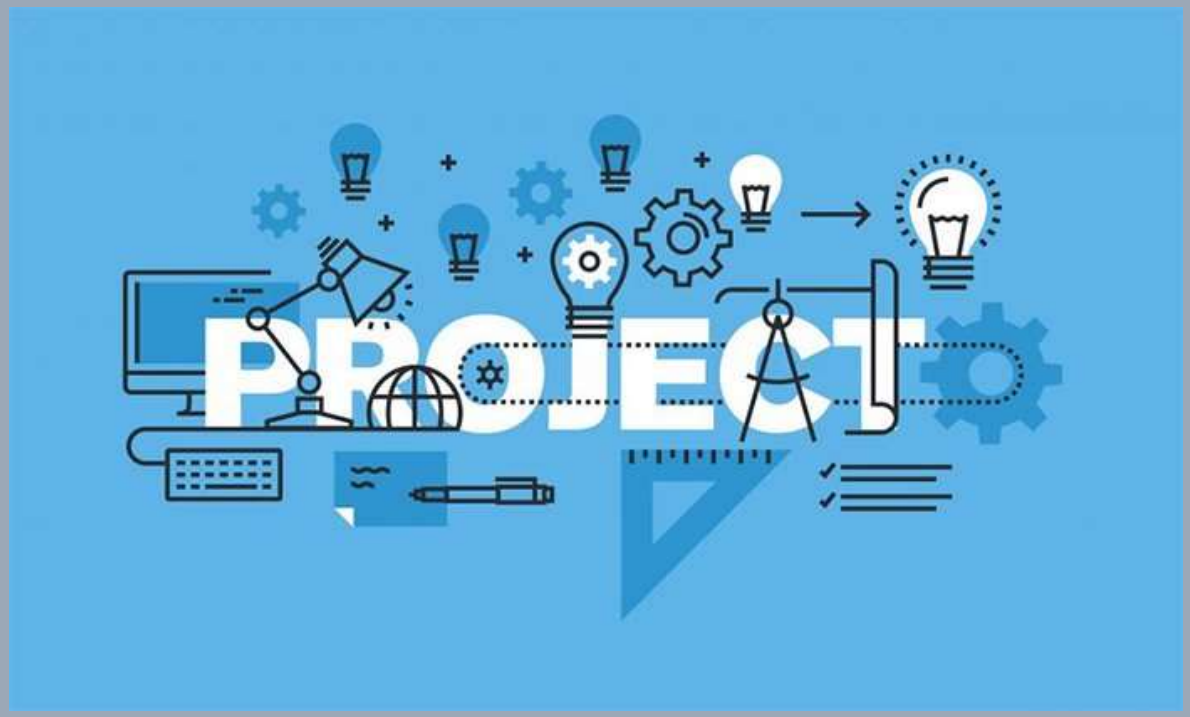
Kospi200 선물 자동매매 System

™ 놀부유희

Content

Content

™ 놀부유희



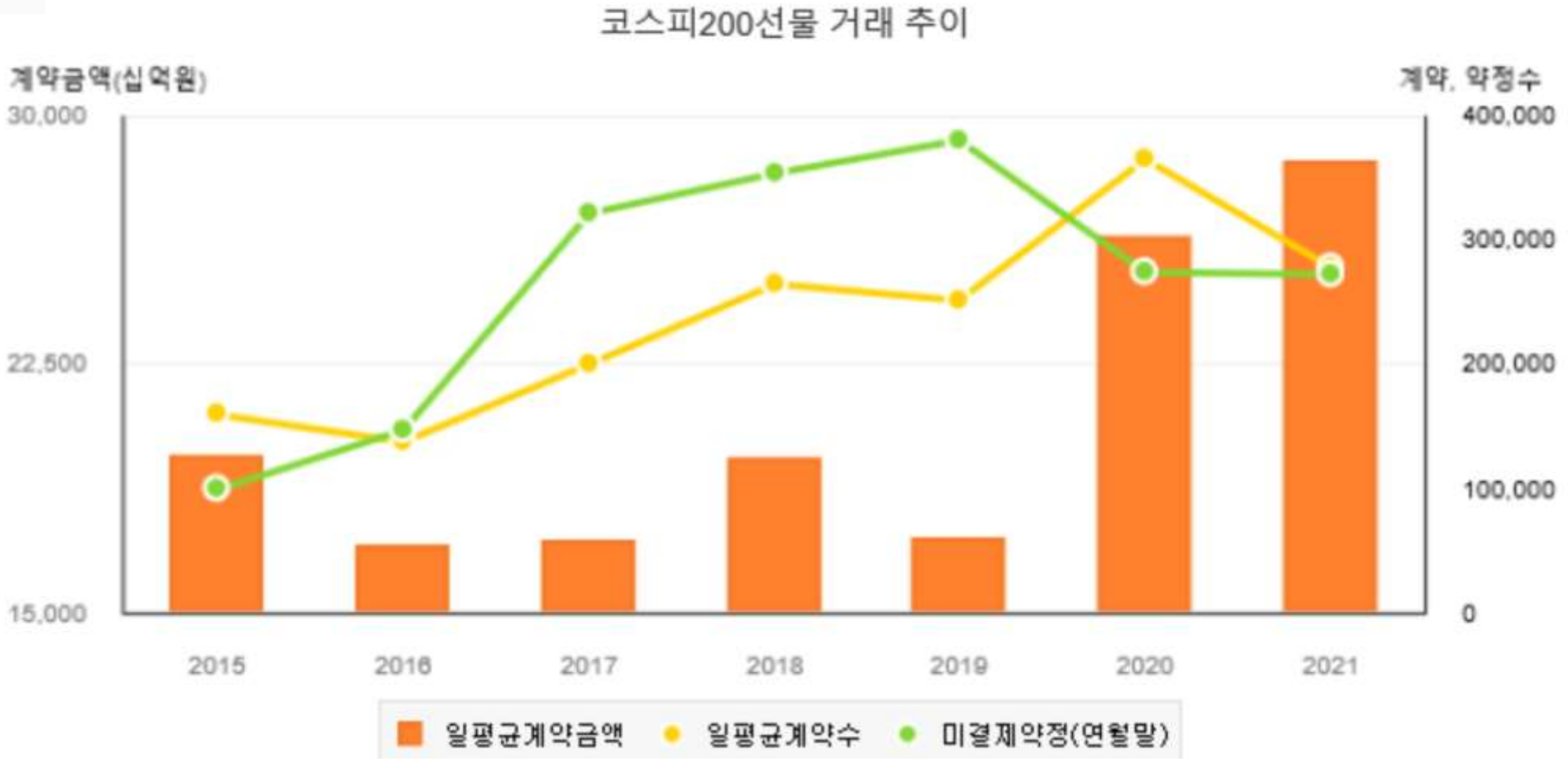
Content

CONTENTS	3월 1주	3월 2주	3월 3주	3월 4주	4월 1주	4월 2주	4월 3주	4월 4주	5월 1주	비고
프로젝트 방향설정	■									
모델선정	■	■								
데이터수집	■	■	■	■	■	■				
논문리뷰		■	■	■	■	■	■	■		
모델구성			■	■	■	■				
매매전략 검증						■	■	■		
테스트 및 개선						■	■	■	■	
최종점검								■	■	

프로젝트 진행 상황 리뷰

CONTENTS	3월 1주	3월 2주	3월 3주	3월 4주	4월 1주	4월 2주	4월 3주	4월 4주	5월 1주	비고
프로젝트 방향설정										
모델선정										
데이터수집										
논문리뷰					Transformer 시작					
모델구성						ARIMA, Prophet, LSTM, XGBoost, TFT				
매매전략 검증					과거 데이터 검증, 모의 테스트					
테스트 및 개선							모델 별 가중치 설정			
최종점검								배포 및 테스트		


Why?


코스피200선물 거래 추이
계약금액(십억원)
계약, 약정수
30,000
22,500
15,000
400,000
300,000
200,000
100,000
0
2015
2016
2017
2018
2019
2020
2021
일평균계약금액
일평균계약수
미결제약정(연월말)

합 격 증

Certificate

성 명 : 이소담
Full Name

생 년 월 일 : 1977. 01. 05
Date of Birth January 05, 1977

자격시험명 : 파생상품투자상담사
Name of Examination Certified Derivatives Investment Advisor

합격증번호 : 14-001039
Certificate Number

KOFIA

위 사람은 한국금융투자협회가 실시한 제20회 파생상품투자상담사 자격시험에 합격하였기에 본 증서를 수여함.

This is to certify that the above mentioned person has passed the 20th Certified Derivatives Investment Advisor Qualification Examination conducted by the Korea Financial Investment Association.

[투자자 유형에 따른 기본예탁금 단계]

투자자유형	거래가능상품	기본예탁금	예탁금단계	비고
전문투자자	모든 거래	기본예탁금 면제, 사전교육 및 모의거래 면제		
기존투자자	모든 거래	500만원	기존투자1	사전교육 및 모의거래 면제
		1,500만원	기존투자2	
		3,000만원	기존투자3	
일반투자자	선물 및 옵션매수	1,000만원 (신규투자자)	선물 / 옵션매수1	[사전교육] - 1시간 : 위험선호형 & 파생상품투자경험 Y - 3시간 : 위험선호형 &파생상품투자경험 N - 10시간 : 부적합투자자, 65세이상고령자 [모의거래] - 3시간 : 위험선호형 & 파생상품투자경험 Y - 5시간 : 위험선호형 & 파생상품투자경험 N - 10시간 : 부적합투자자, 65세이상고령자 * 옵션매도 및 변동성지수선물거래 계좌개설 후 미결제약정10거래일 이상보유
		1,500만원	선물 / 옵션매수2	
	모든 거래	2,000만원 (신규투자자)	옵션매도1	
		2,500만원	옵션매도2	

KOSPI200 주가지수 선물/옵션 매매제도

증거금

위탁증거금률	유지증거금률
10.20%	6.80%

최소가격변동금액	12,500원 (25만원×0.05)
거래시간	09:00 ~ 15:45 (최종거래일 09:00 ~ 15:20)
최종거래일	각 결제월의 두 번째 목요일(공휴일인 경우 순차적으로 앞당김)
결제월	3, 6, 9, 12월, 2년 반기월물(6월) 1개, 3년 12월물 2개 총 7개 종목

종목코드	종목명	기준가	증거금률(%)	승수	계약당 증거금
101S6000	KOSPI200 F 202206	365.61	10.20	250,000.00000000	9,323,055

미니 코스피200선물/옵션 매매제도

구분	미니코스피200선물	미니코스피200옵션
기초자산	코스피200지수	
거래승수	5만원	
결제월수	연속월6개 (각 결제월물의 거래기간은 6개월)	

상품구분 지수선물 | 선물/스프레드 구분 선물 | 종목 05S4000 | 조회 | 다음

종목코드	종목명	기준가	증거금율(%)	승수	계약당 증거금
105S4000	미니KOSPI F 202204	365.61	10.20	50,000.00000000	1,864,611

288.7 – 285.5 = 3.2

3.2 / 0.05 = 64 틱

64틱 * 12,500 원
= 800,000 원

800,000원 x 5 계약
= 4,000,000 원

필요증거금
46,615,275 원

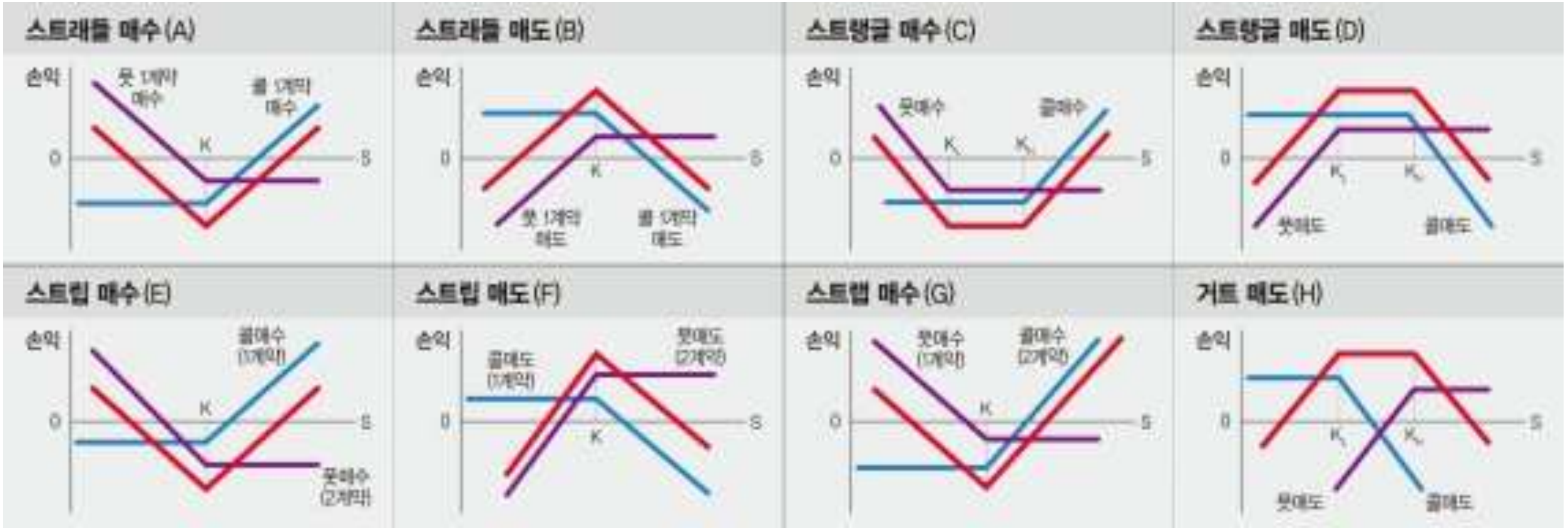
수익률 + 8.58 %

손익계산

진입	청산	실현손익
매수	매도	
285.50	288.70	4,000,000원 이익
< 5계약 >	< 5계약 >	

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	날짜	시가	고가	저가	종가		고가-저가						
2	2022/04/01	364.1	364.5	361.5	363.5		3.00						
3	2022/03/31	364.75	367.75	364.45	365.7		3.30						
4	2022/03/30	367	367.5	364.3	364.75		3.20			기간	포인트	틱	1 계약 기대수익
5	2022/03/29	364.3	365.45	362.25	363.55		3.20		1일	5일	3.18	63.5	₩793,750
6	2022/03/28	359.55	362.85	358.3	362.45		4.55		변동성 평균	20일	3.80	76	₩950,000
7	2022/03/25	363.5	364.1	360.75	361.75		3.35		(고가-저가)	60일	3.83	76.5	₩956,250
8	2022/03/24	361.5	363.2	360.15	362.5		3.05			120일	6.63	132.5	₩1,656,250
9	2022/03/23	363.6	366.4	362.7	364.25		3.70			200일	3.75	75	₩937,500
10	2022/03/22	358.4	361.95	358.2	361.4		3.75						
11	2022/03/21	363.15	363.75	357.9	358.75		5.85				4.235	84.7	₩1,058,750
12	2022/03/18	360.15	362.45	359.2	362		3.25						
13	2022/03/17	362.95	364.15	360.7	361.75		3.45						
14	2022/03/16	354	356.2	351.45	356.1		4.75						
15	2022/03/15	352.2	353.05	349.3	350.3		3.75						

高 변동성!
예측 할 수 없는
내일!



www.saramin.co.kr › zf_user

(주)**SD경제**연구소 - 자산운용/**파생**상품트레이딩/선물옵션트레이딩/선물옵션....

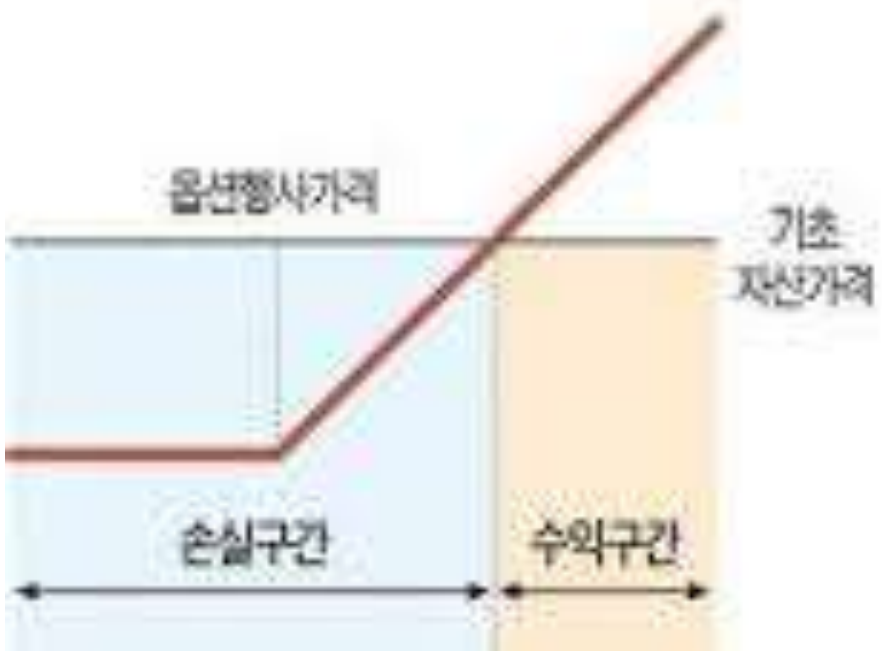
(주)**SD경제**연구소 채용 정보, 담당업무: 자산운용/**파생**상품트레이딩/선물옵션트레이딩/선물옵션매매/**파생**상품 운용, 근무지역: 전국, 인근지하철: 서울 5호선 여의도, 근무형태: 프리랜서, 모집인원: 00명, 학력: 학력무관, 경력: 경력무관, 연봉: 회사내규에 따름, , 직급/직책: 면접후 결정 실장 근무부서: 운용팀, 근무위치: 재택근무, 직...

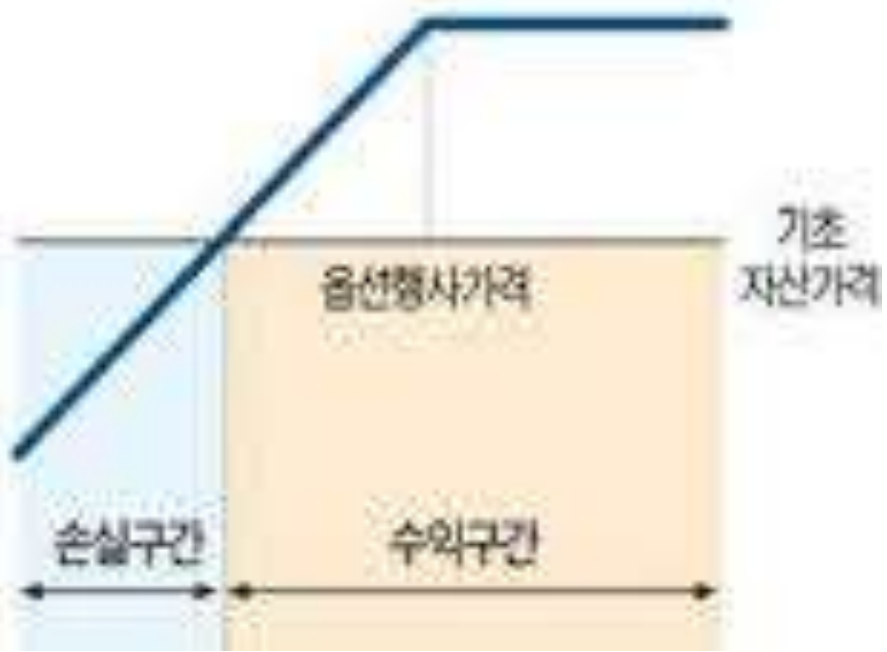
2015.01.18.

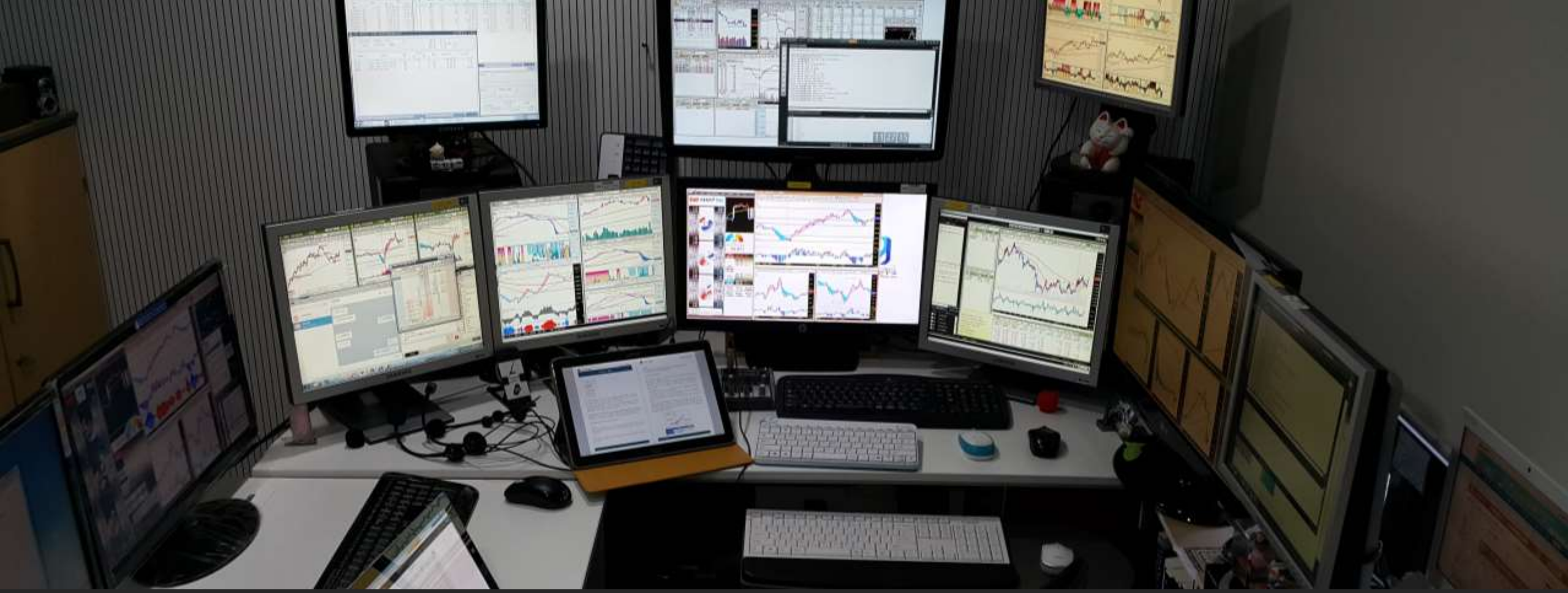
콜옵션 매수시 수익곡선

풋옵션 매도시 수익곡선

손익 관리의 중요성 !







2012 증권사 재직 시절

이동평균 교차 시스템 전략

MACD돌파 시스템 전략

추세선 자동 인디케이터 시스템 전략

추세추종자 시스템 전략

헤드앤숄더 시스템 전략

반락 매수 . 랠리 매도 시스템 전략

DMI 시스템 전략

켈트너 채널 시스템 전략

선형회귀 & 모멘텀 시스템 전략 I

선형회귀 & 모멘텀 시스템 전략 II

SignalMaker

TradeStation®

DMA채널 & 레인지 리더 시스템 전략

미분 - 이동 평균 시스템 전략

Ryan의 HOPE 트레이딩 전략

High Five 시스템 전략

Hit the Bull's Eye 시스템 전략

Dr Tharf의 샹들리에 시스템 전략

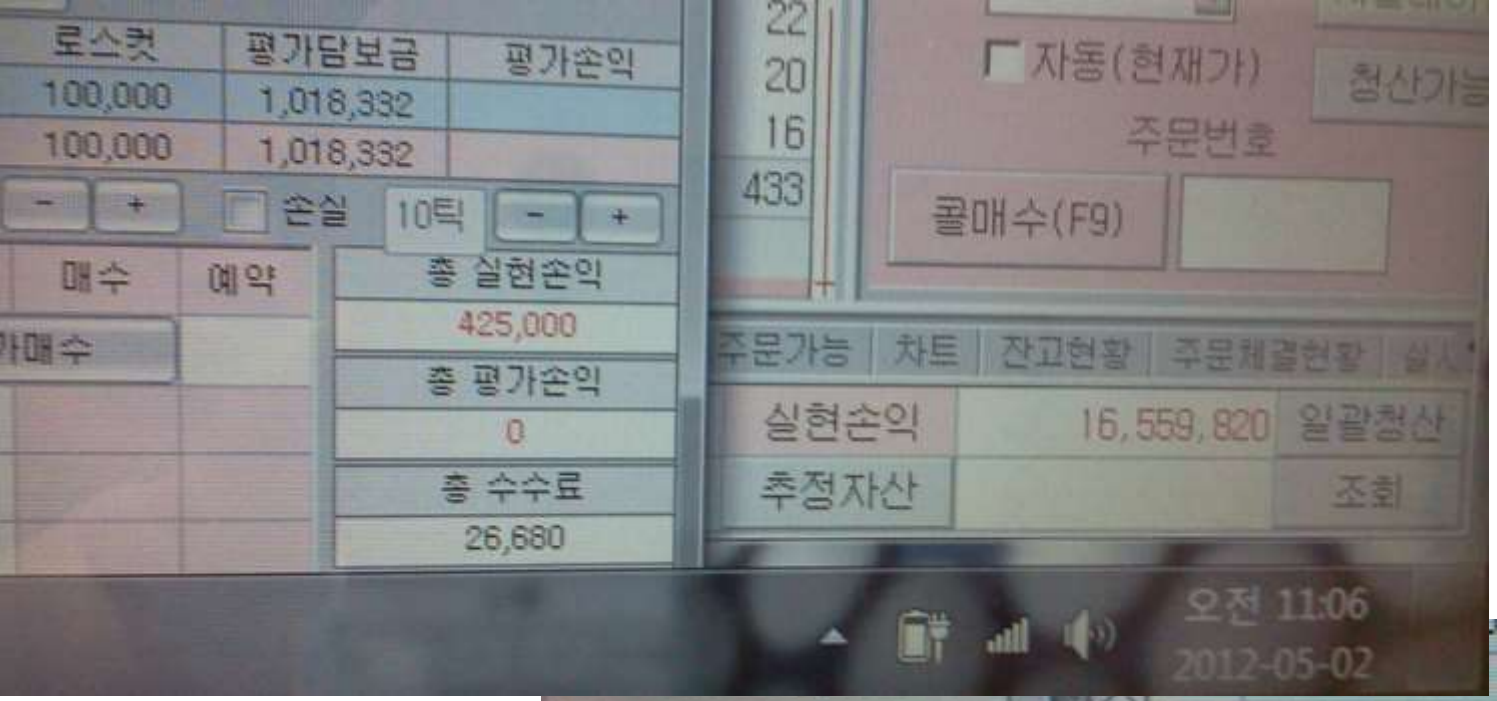
Joe Krutsinger의 야간열차 시스템 전략

Joe Stowell의 In The Zone 시스템 전략

Ricahrd Wyckoff의 Spring and Upthrust 전략

Three Musketeers 시스템 전략

Walter Downs의 Tiger in the mist전략


로스컷
평가담보금
평가손익
100,000
1,018,332
100,000
1,018,332
손실
10틱
매수
예약
총 실현손익
425,000
총 평가손익
0
총 수수료
26,680
22
20
16
433
자동(현재가)
주문번호
콜매수(F9)
주문가능
차트
잔고현황
실현손익
16,559,820
일괄청산
추정자산
조회
오전 11:06
2012-05-02

+α


3
풋매수(F9)
700,000
총 평가손익
0
총 수수료
60,046
수량
기간손익
포지션
예탁금
주문가능
차트
잔고현황
총손익
559,180
실현손익
19,895,390
수익률
0.31%
추정자산
오후 2:30
2012-05-02

KOSPI 200지수선물 / 미니지수 선물

데이 트레이딩 (포지션 오버나잇 X)

철저한 손익 관리

고객 진입 시그널

(계좌 주문 연계)

진입 시점 분석

방향성 분석 (Classification)

목표 가격 분석 (Regression)

② ① ③ ⑤ ⑥

회원가입
회원관리

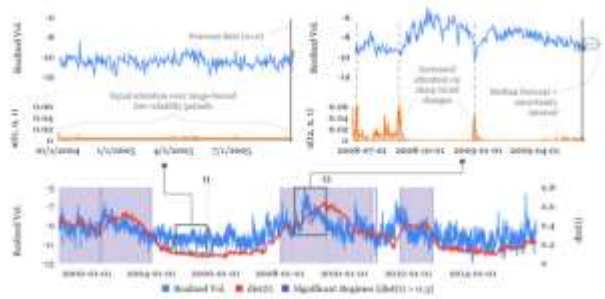
④

(프론트 엔드 연계)

④

ForeCast Model

ARIMA
Prophet
XG Boost
LSTM
Temporal Fusion Transformer



① ② ③ ④ ④ ⑤ ⑥

회원가입
회원관리

(프론트 엔드 연계)

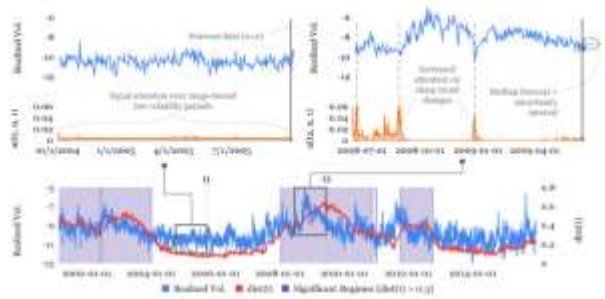
ForeCast Model

ARIMA

Prophet

XG Boost

LSTM



Temporal Fusion Transformer

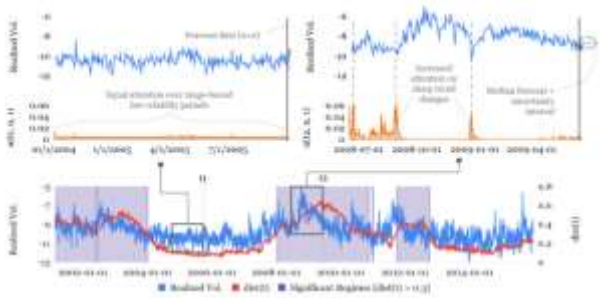
ForeCast Model

ARIMA

Prophet

XG Boost

LSTM



Temporal Fusion Transformer

When

FINDING
LONG BODY

CNN ?
DNN ?
엑셀 ?

Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market

Rosdyana Mangir Irawan Kusuma[1], Trang-Thi Ho[2], Wei-Chun Kao[3], Yu-Yen Ou[1] and Kai-Lung Hua[2]

[1]Department of Computer Science and Engineering, Yuan Ze University, Taiwan Roc
[2]Department of Computer Science and Engineering, National Taiwan University of Science and Technology, Taiwan Roc
[3]Omniscient Cloud Technology

4.2.2 Residual Network

It is an artificial neural network developed by He in 2015 [4]. It uses skip connections or shortcut to jump over some layers. The key of residual network architecture is the residual block, which allows information to be passed directly through. Therefore, the backpropagated error signals is reduced or removed. This allows to train a deeper network with hundreds of layers. And this vastly increased depth led to significant performance achives.

4.2.3 VGG Network

The VGG network architecture was introduced by Simonyan and Zisserman[14]. It is named VGG because this architecture is from VGG group, Oxford. This network is characterized by its simplicity, using only 3x3 convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. Two fully connected layers, each with 4096 nodes are then followed by a softmax classifier. The 16 and 19 stand for the number of weight layers in the network. Unfortunately, there are two major drawbacks with VGGNet. First, it is painfully slow to train and the second the network architecture weights themselves are quite large.

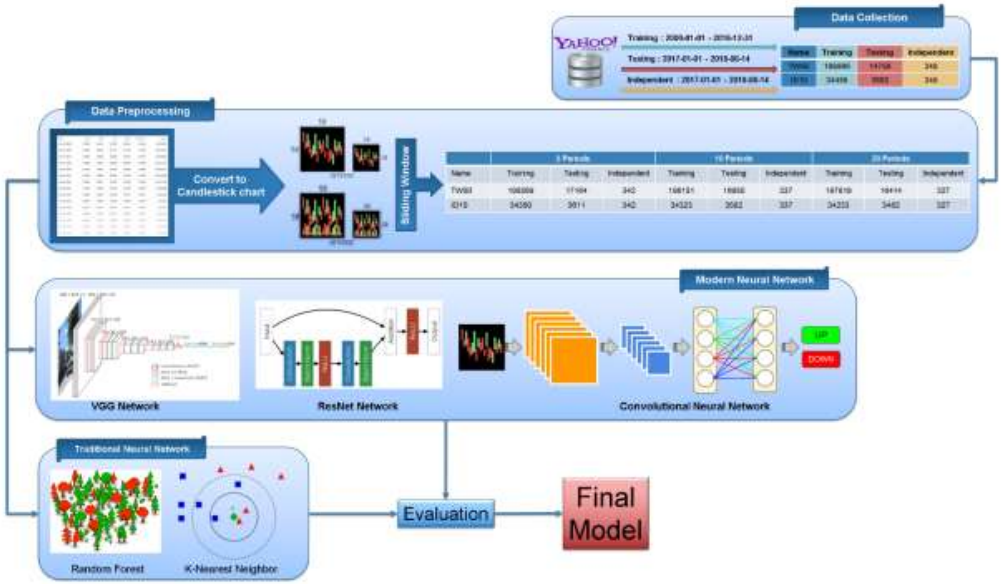
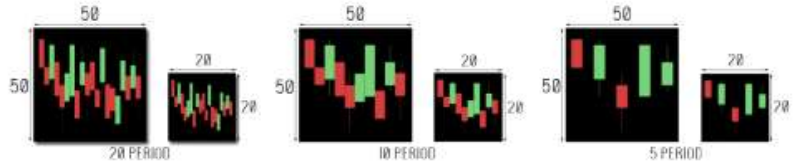
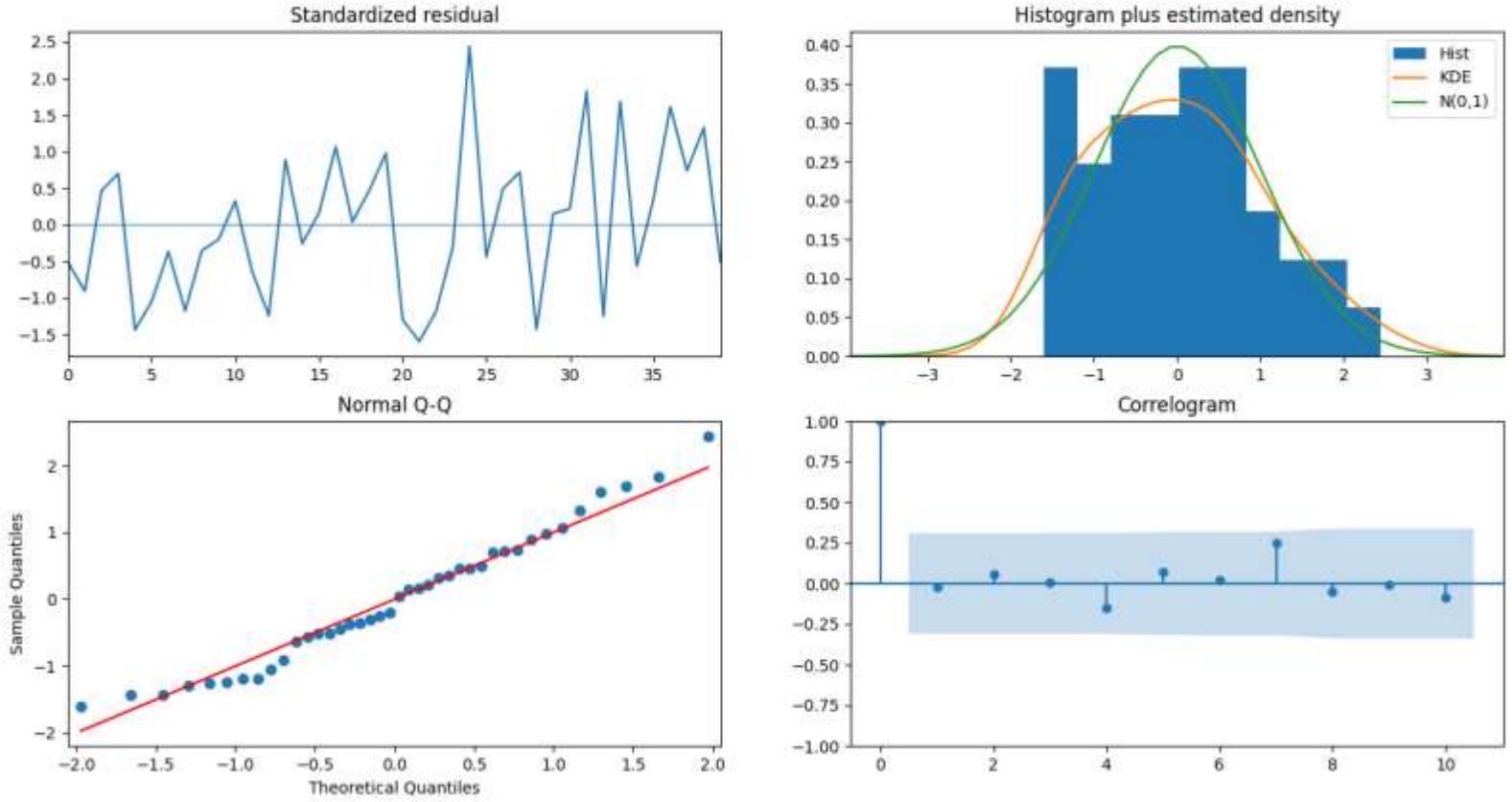


Figure 1: Our methodology design.



Content

ARIMA


Standardized residual
Histogram plus estimated density
Hist
KDE
N(0,1)
Normal Q-Q
Sample Quantiles
Theoretical Quantiles
Correlogram

ARIMA (Autoregressive Integrated Moving Average) 응용전략

API를 활용한 스캘핑 매매기법
→ 1분봉 Data 종가 활용 (1일 거래시간 396분)

1주(5일간), 1달(20일), 1분기(60일)
→ 기간별 ARIMA 모형 도출, 최빈값 모형으로 예측

최적의 진입 시간대 (추세 구간)
ARIMA 를 활용한 예측

ARIMA
(Autoregressive Integrated Moving Average)

자기회귀와 이동평균을 동시 활용

비정상성(Non-stationary)을 설명하기 위해 관측치간의 차분(Diffrance)을 사용

1차 차분: $Y_t = X_t - X_{t-1} = \nabla X_t$

2차 차분: $Y_t^{(2)} = X_t - X_{t-2} = \nabla^{(2)} X_t$

d차 차분: $Y_t^{(d)} = X_t - X_{t-d} = \nabla^{(d)} X_t$

X
2
7
10
5
8

−

X
2
7
10
5
8

=

Y
5
3
-5
3
-

- AR(p) = ARIMA(p, 0, 0)
- MA(q) = ARIMA(0, 0, q)
- ARMA(p, q) = ARIMA(p, 0, q)

ARIMA
(Autoregressive Integrated Moving Average)

AR모형 Lag → p

차분(Diffrence)횟수 → d

MA모형 Lag → q

ACF plot

$$\rho_k = \frac{Cov(y_t, y_{t+k})}{Var(y_t)}$$

ACF(Autocorrelation function) :
Lag에 따른 관측치들 사이의 관련성을 측정하는 함수

PACF plot

PACF(Partial autocorrelation function) :
k 이외의 모든 다른 시점 관측치의 영향력을 배제하고 y_t와 y_{t-k} 두 관측치의 관련성을 측정하는 함

$$\phi_{kk} = corr(y_t, y_{t-k} \mid y_{t-1}, y_{t-2}, \cdots, y_{t-k+1})$$

통상적

p + q < 2

p * q = 0

대부분의 시계열 자료에서는 하나의 경향만이 강하다

ARIMA
(Autoregressive Integrated Moving Average)

ACF plot

ACF(Autocorrelation function) :
Lag에 따른 관측치들 사이의 관련성을 측정하는 함수

시계열 데이터가 AR의 특성을 띄는 경우
ACF는 천천히 감소하고 PACF는 처음 시차를 제외하고 급격히 감소

PACF plot

PACF(Partial autocorrelation function) :
k 이외의 모든 다른 시점 관측치의 영향력을 배제하고 y_t 와 y_{t-k} 두 관측치의 관련성을 측정하는 함

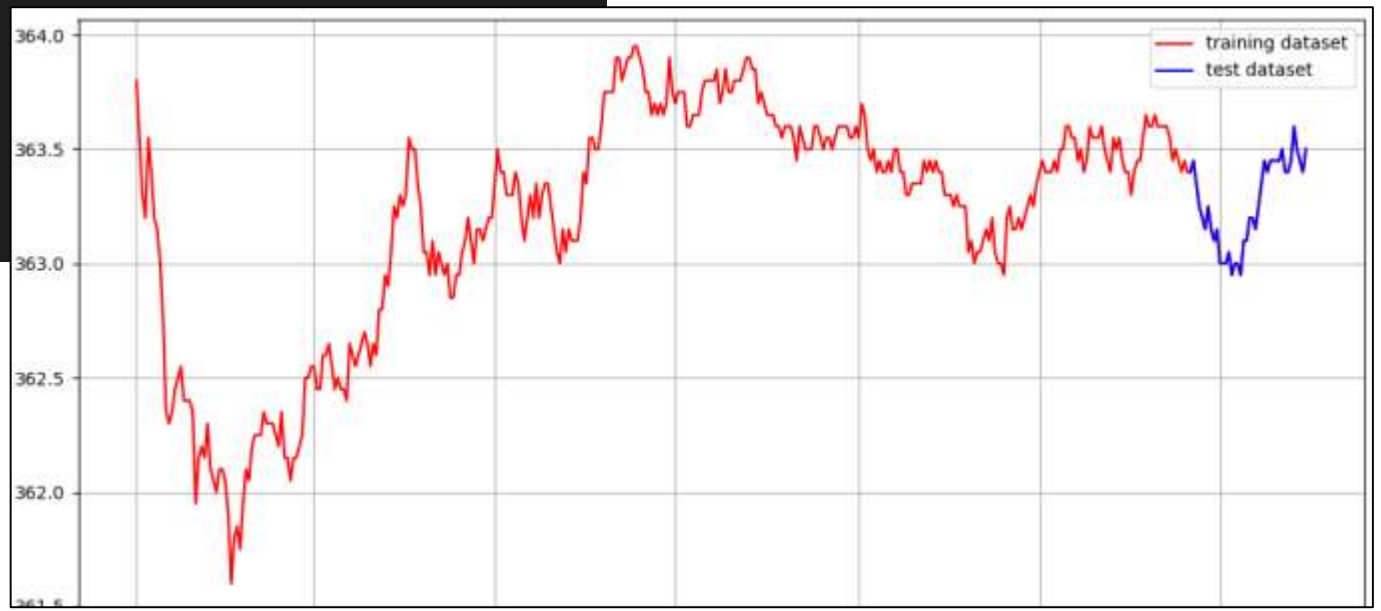
MA의 특성을 띄는 경우
ACF는 급격히 감소하고 PACF는 천천히 감소한다.

급격히 감소하는 시차를 각 AR과 MA 모형의 모수(p, q)로 사용할 수 있다.
또한 데이터를 d차분하여
ACF 및 PACF 계산함으로써 적절한 차분횟수까지 구할 수 있다

Statistical forecasting:
notes on regression and time series analysis

Robert Nau
Fuqua School of Business
Duke University

```python
# 하루치 1분봉 시계열 종가 데이터 준비
train_data, test_data = ts_log[:int(len(ts_log)*0.9)], ts_log[int(len(ts_log)*0.9):]

plt.figure(figsize=(13,6))
plt.grid(True)
plt.plot(ts_log, c='r', label='training dataset')
plt.plot(test_data, c='b', label='test dataset')
plt.legend()
plt.show()
```



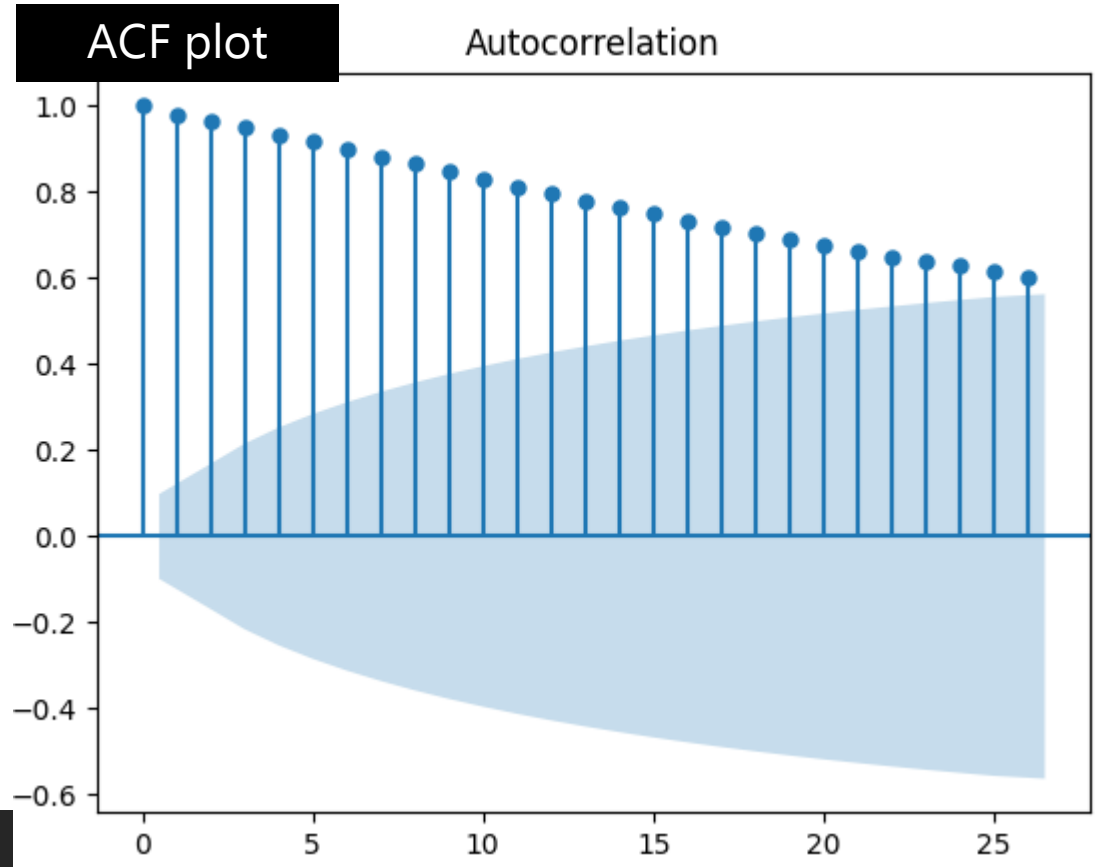
```python
# ACF, PACF 그려보기 -> p,q 구하기
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

plot_acf(ts_log)    # ACF : Autocorrelation 그래프 그리기
plot_pacf(ts_log)  # PACF : Partial Autocorrelation 그래프 그리기
plt.show()

# 차분 안정성 확인 -> d 구하기
# 1차 차분 구하기
print('\033[31m'+'\033[1m' + "1차 차분 구하기 :" + '\033[0m')
diff_1 = ts_log.diff(periods=1).iloc[1:]
```

시계열 데이터가 AR의 특성을 띄는 경우

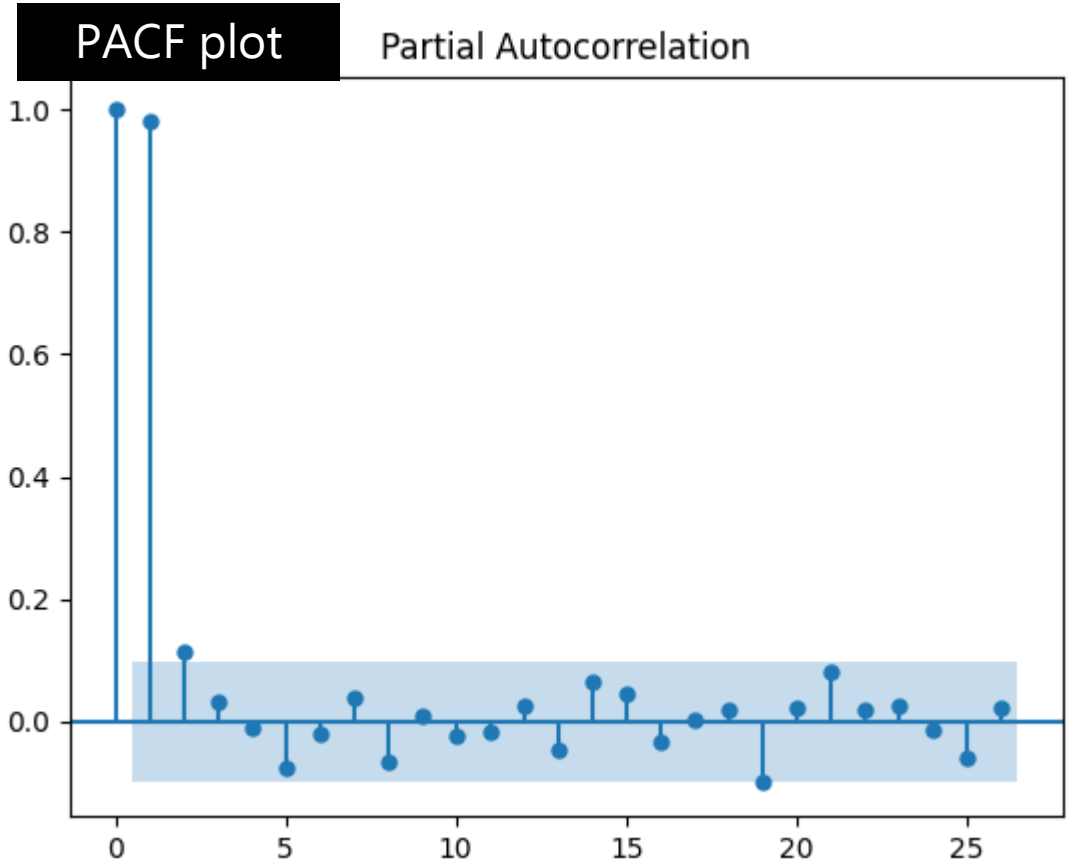
ACF는 천천히 감소하고
PACF는 처음 시차를 제외하고 급격히 감소

p = 0, q = 1

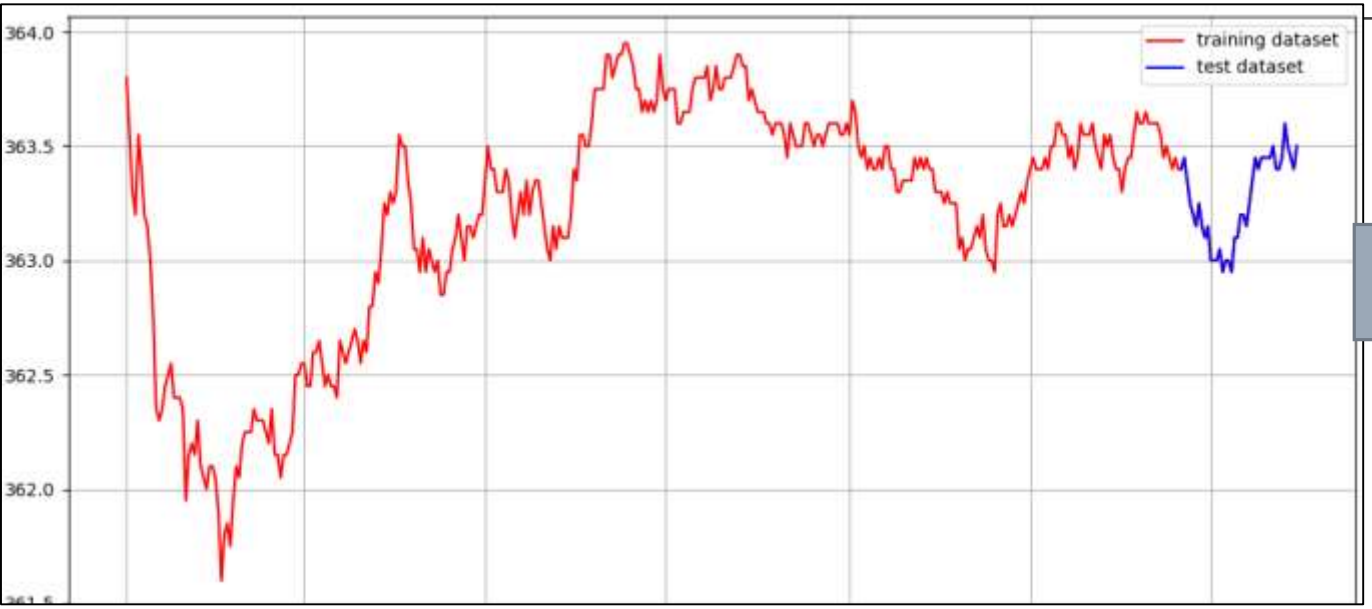
ACF plot

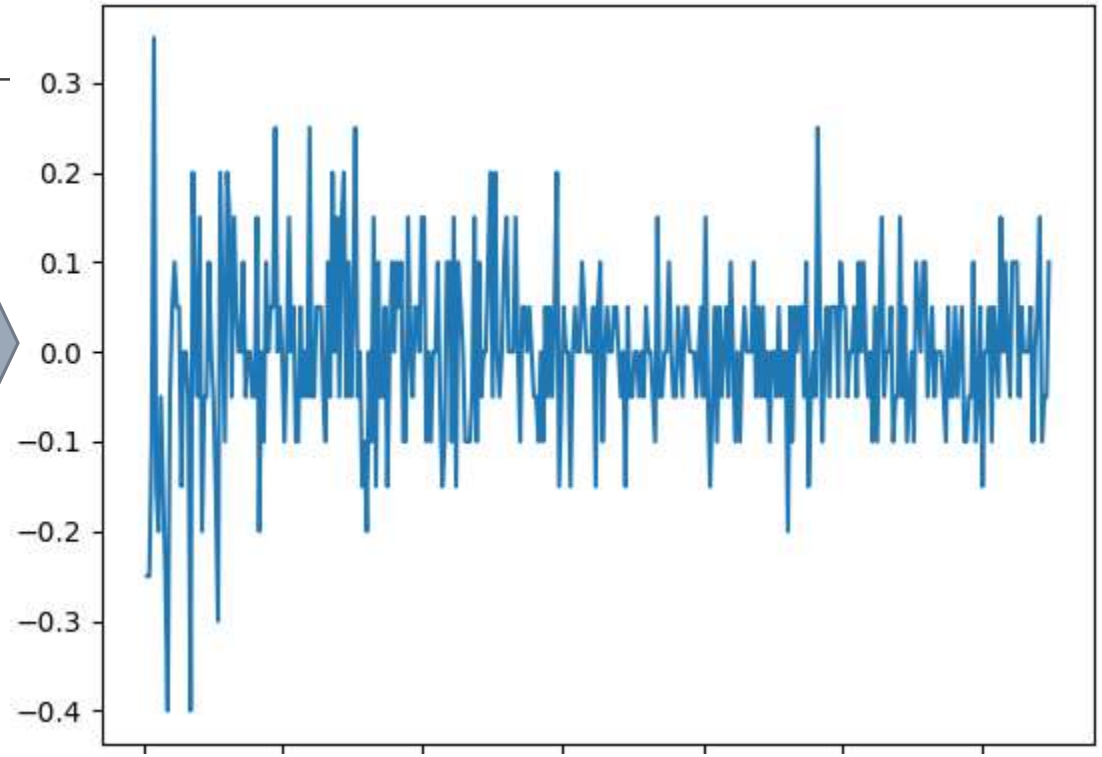


PACF plot



```
diff_1 = ts_log.diff(periods=1).iloc[1:]
plt.plot(diff_1)
plt.show()
plot_acf(diff_1)   # ACF : Autocorrelation 그래프 그리기
plot_pacf(diff_1)  # PACF : Partial Autocorrelation 그래프 그리기
```

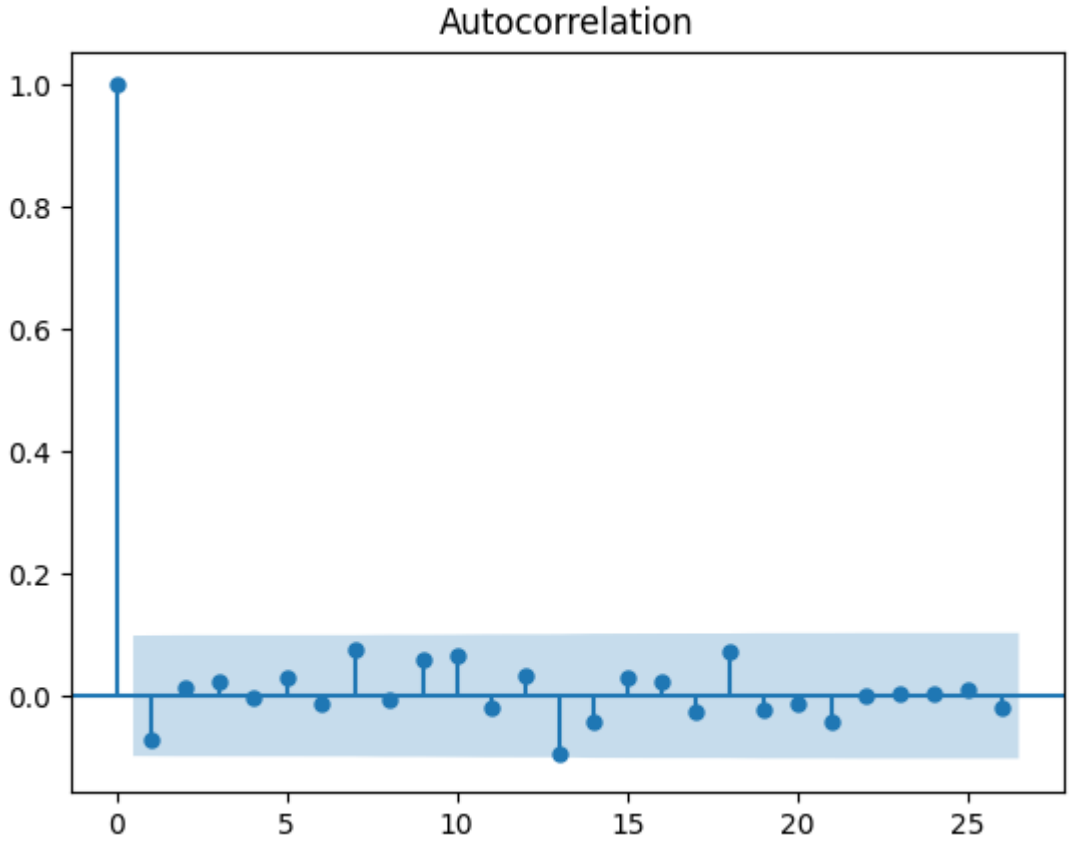


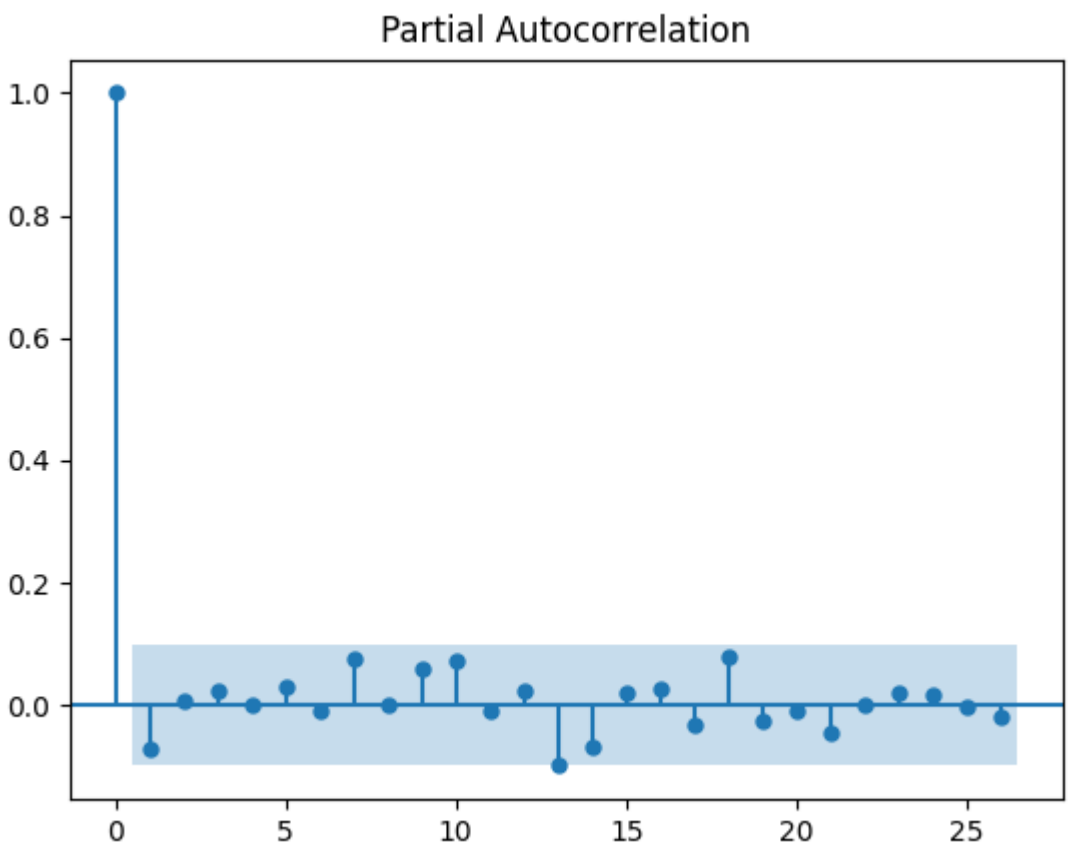


1 차분이후 시계열 정상상태(Stationary)

1 차분이후의 ACF와 PACF
시계열 정상상태(Stationary)

ARIMA(p, d, q) → ARIMA(0, 1, 1)





R에서 상수를 이해하기

비-계절성 ARIMA 모델을 다음과 같이 쓸 수 있습니다.

$$(1-\phi_1 B-\cdots-\phi_p B^p)(1-B)^d y_t = c+(1+\theta_1 B+\cdots+\theta_q B^q)\varepsilon_t, \qquad (8.4)$$

또는 다음의 것과 동일하게

$$(1-\phi_1 B-\cdots-\phi_p B^p)(1-B)^d (y_t-\mu t^d/d!) = (1+\theta_1 B+\cdots+\theta_q B^q)\varepsilon_t, \qquad (8.5)$$

여기에서 $c=\mu(1-\phi_1-\cdots-\phi_p)$이고 μ는 $(1-B)^d y_t$의 평균입니다. R에서는 매개변수화된 식 (8.5)을 사용합니다.

따라서, 정상성을 나타내지 않는 ARIMA 모델에서 상수가 포함되는 것은 예측 함수에서 차수 d의 다항식 추세를 유도하는 것과 같습니다. (상수가 생략되면, 예측 함수는 차수 $d-1$의 다항식 추세를 포함합니다.) $d=0$일 때는, μ가 y_t의 평균인 특수한 상황이 됩니다.

기본값으로, `Arima()` 함수는 $d>0$일 때 $c=\mu=0$로 정하고 $d=0$일 때 μ의 추정치를 냅니다. 이 값은 시계열의 표본 평균에 가까워질 것이지만, $p+q>0$일 때, 표본 평균이 최대 가능도 추정(maximum likelihood estimation)이 아니기 때문에 보통은 정확하게 같지는 않습니다.

입력값 `include.mean` 은 $d=0$일 때만 영향을 주고 기본값은 `TRUE` 입니다. `include.mean=FALSE` 로 두면 강제로 $\mu=c=0$로 둘 것입니다.

입력값 `include.drift` 은 $d=1$일 때 $\mu\neq 0$을 허용합니다. $d>1$일 때는 2차나 이상의 차수 추세가 예측할 때 특별히 위험할 수 있기 때문에 어떠한 상수도 허용하지 않습니다. $d=1$일 때 매개변수 μ는 R 출력에서 "표류(drift)"라고 부릅니다.

`include.constant` 라는 입력값도 있는데 `TRUE` 이면 $d=0$일 때 `include.mean=TRUE` 로 둘 것이고, $d=1$ 일 때는 `include.drift=TRUE` 로 둡니다. `include.constant=FALSE` 이면, `include.mean` 와 `include.drift` 둘 다 `FALSE` 로 두게 됩니다. `include.constant` 을 사용하면, `include.mean=TRUE` 와 `include.drift=TRUE` 는 무시됩니다.

`auto.arima()` 함수는 상수를 포함하는 과정을 자동화합니다. 기본값으로, $d=0$나 $d=1$에 대해, AICc 값이 좋아지면 상수가 추가될 것입니다. $d>1$에 대해, 상수가 항상 생략됩니다. `allowdrift=FALSE` 로 정하면, $d=0$일 때만 상수가 허용됩니다.

```python
# ARIMA 모델 훈련
from statsmodels.tsa.arima_model import ARIMA
# Build and Train  Model (p, d, q)
model = ARIMA(train_data, order=(0, 1, 1))
fitted_m = model.fit(trend='c',full_output=True, disp=True)
'''
disp : bool, optional
If True, convergence information is print
For the default l_bfgs_b solver,
disp controls the frequency of the output
disp < 0 means no output in this case.
'''
print(fitted_m.summary())
```

```
CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
                             ARIMA Model Results
==============================================================================
Dep. Variable:                 D.Close   No. Observations:                  355
Model:                 ARIMA(0, 1, 1)   Log Likelihood                 329.650
Method:                       css-mle   S.D. of innovations              0.096
Date:                Mon, 04 Apr 2022   AIC                           -653.301
Time:                        01:03:14   BIC                           -641.684
Sample:                    01-02-2020   HQIC                          -648.679
                         - 12-21-2020
==================================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
const             -0.0011      0.005     -0.226      0.822      -0.010       0.008
ma.L1.D.Close     -0.0670      0.053     -1.273      0.203      -0.170       0.036
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1           14.9340           +0.0000j           14.9340            0.0000
-----------------------------------------------------------------------------
```

Const P > | z | : 0.822

→ 5% 수준 기각 X

학습의 적정성을 위해 확인되는
t-test값
p value 0.05수준에서 보면
MA(1)의 계수는 유효
Constant는 유효하지 않다

```python
# ARIMA 모델 훈련
from statsmodels.tsa.arima_model import ARIMA
# Build and Train  Model (p, d, q)
model = ARIMA(train_data, order=(0, 1, 1))
fitted_m = model.fit(trend='nc',full_output=True, disp=True)  #상수항 사용 X
'''
disp : bool, optional
If True, convergence information is printe
For the default l_bfgs_b solver,
disp controls the frequency of the output
disp < 0 means no output in this case.
'''
print(fitted_m.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                 D.Close   No. Observations:                  355
Model:                 ARIMA(0, 1, 1)   Log Likelihood                 329.625
Method:                       css-mle   S.D. of innovations              0.096
Date:                Mon, 04 Apr 2022   AIC                           -655.250
Time:                        01:15:47   BIC                           -647.505
Sample:                    01-02-2020   HQIC                          -652.169
                         - 12-21-2020
==================================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
ma.L1.D.Close     -0.0669      0.053     -1.272      0.203      -0.170       0.036
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1           14.9405           +0.0000j           14.9405            0.0000
-----------------------------------------------------------------------------
```

Const P > | z | : 0.203

→ 5% 수준 기각 X

학습의 적정성을 위해 확인되는
t-test값
p value 0.05수준에서 보면
MA(1)의 계수는 유효
Constant는 유효하지 않다

Auto ARIMA

```
from pmdarima.arima import ndiffs
import pmdarima as pm

kpss_diffs = ndiffs(ts_log, alpha=0.05, test='kpss', max_d=6)
adf_diffs = ndiffs(ts_log, alpha=0.05, test='adf', max_d=6)
n_diffs = max(adf_diffs, kpss_diffs)
print(f"추정된 차수 d = {n_diffs}")
best_model = pm.auto_arima(y = ts_log
                           , d = 1
                           , start_p = 0 #(기본값 = 2), max_p (기본값 = 5): AR(p)를 찾을 범위 (start_p에서 max_p까지 찾는다!)
                           , max_p = 3
                           , start_q = 0 # (기본값 = 2), max_q (기본값 = 5): AR(q)를 찾을 범위 (start_q에서 max_q까지 찾는다!)
                           , max_q = 3
                           , m = 1  #(기본값 = 1): 계절적 차분이 필요할 때 쓸 수 있는 모수로
                                    # m=4이면 분기별, m=12면 월별, m=1이면 계절적 특징을 띠지 않는 데이터를 의미
                                    # m=1이면 자동적으로 seasonal 에 대한 옵션은 False로 지정.
                           , seasonal = False #(기본값 = True): 계절성 ARIMA 모형을 적합할지의 여부
                           , stepwise = True #(기본값 = True): 최적의 모수를 찾기 위해 쓰는 힌드만 - 칸다카르 알고리즘을 사용할지의 여부
                                             #False면 모든 모수 조합으로 모형을 적합한다.
                           , trace=True #(기본값 = False): stepwise로 모델을 적합할 때마다 결과를 프린트하고 싶을 때 사용한다.
                           )
print("best model --> (p, d, q):", best_model.order)
```

```
추정된 차수 d = 1
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-743.579, Time=0.04 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-743.616, Time=0.05 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-743.572, Time=0.12 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-745.554, Time=0.02 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-741.627, Time=0.21 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.441 seconds
best model --> (p, d, q): (0, 1, 0)
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  396
Model:               SARIMAX(0, 1, 0)   Log Likelihood                 373.777
Date:                Mon, 04 Apr 2022   AIC                           -745.554
Time:                        01:38:48   BIC                           -741.575
Sample:                             0   HQIC                          -743.977
                                  - 396
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2         0.0088      0.000     19.561      0.000       0.008       0.010
===================================================================================
Ljung-Box (L1) (Q):                   2.01   Jarque-Bera (JB):                59.39
Prob(Q):                              0.16   Prob(JB):                         0.00
Heteroskedasticity (H):               0.33   Skew:                            -0.17
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.87
===================================================================================
```

1차 차분을 했을 때 백색 잡음 ($\epsilon_t \sim N(0,\sigma_2)$ $\epsilon t \sim N(0,\sigma 2)$)
(정상성을 만족)

임의 보행 모형 (Random Walk Model)을 따른다

잔차가 백색잡음 과정인지
(=정상성을 만족하는지),
정규성 및 등분산성을 만족하는지 파악

```
추정된 차수 d = 1
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-743.579, Time=0.04 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-743.616, Time=0.05 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-743.572, Time=0.12 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-745.554, Time=0.02 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-741.627, Time=0.21 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.441 seconds
best model --> (p, d, q): (0, 1, 0)
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  396
Model:               SARIMAX(0, 1, 0)   Log Likelihood                 373.777
Date:                Mon, 04 Apr 2022   AIC                           -745.554
Time:                        01:38:48   BIC                           -741.575
Sample:                             0   HQIC                          -743.977
                                  - 396
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2         0.0088      0.000     19.561      0.000       0.008       0.010
===================================================================================
Ljung-Box (L1) (Q):                   2.01   Jarque-Bera (JB):                59.39
Prob(Q):                              0.16   Prob(JB):                         0.00
Heteroskedasticity (H):               0.33   Skew:                            -0.17
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.87
===================================================================================
```

잔차가 백색잡음 과정인지
정규성 및 등분산성을 만족하는지 파악

•Ljung-Box (Q) 융-박스 검정 통계량
잔차가 백색잡음인지 검정한 통계량

•Prob (Q) 값 0.16
유의수준 0.05에서 귀무가설을 기각하지 못합니다.

Ljung-Box (Q) 통계량의 귀무가설
"잔차(residual)가 백색잡음(white noise)
시계열을 따른다"

위 결과를 통해
시계열 모형이 잘 적합 되었고
남은 잔차는 더이상 자기상관을 가지지 않는
백색 잡음임을 확인

```
추정된 차수 d = 1
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-743.579, Time=0.04 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-743.616, Time=0.05 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-743.572, Time=0.12 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-745.554, Time=0.02 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-741.627, Time=0.21 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.441 seconds
best model --> (p, d, q): (0, 1, 0)
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  396
Model:               SARIMAX(0, 1, 0)   Log Likelihood                 373.777
Date:                Mon, 04 Apr 2022   AIC                           -745.554
Time:                        01:38:48   BIC                           -741.575
Sample:                             0   HQIC                          -743.977
                                  - 396
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2         0.0088      0.000     19.561      0.000       0.008       0.010
===================================================================================
Ljung-Box (L1) (Q):                   2.01   Jarque-Bera (JB):                59.39
Prob(Q):                              0.16   Prob(JB):                         0.00
Heteroskedasticity (H):               0.33   Skew:                            -0.17
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.87
===================================================================================
```

잔차가 백색잡음 과정인지
정규성 및 등분산성을 만족하는지 파악

- Jarque-Bera (JB) 자크-베라 검정 통계량
  잔차가 정규성을 띠는지 검정한 통계량

- Prob(JB)값 0.00
  유의 수준 0.05에서 귀무가설을 기각

Jarque-Bera (JB) 통계량의 귀무가설
"잔차가 정규성을 만족한다"

"잔차가 정규성을 따르지 않음"을 확인

```
추정된 차수 d = 1
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-743.579, Time=0.04 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-743.616, Time=0.05 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-743.572, Time=0.12 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-745.554, Time=0.02 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-741.627, Time=0.21 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.441 seconds
best model --> (p, d, q): (0, 1, 0)
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  396
Model:               SARIMAX(0, 1, 0)   Log Likelihood                 373.777
Date:                Mon, 04 Apr 2022   AIC                           -745.554
Time:                        01:38:48   BIC                           -741.575
Sample:                             0   HQIC                          -743.977
                                  - 396
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2         0.0088      0.000     19.561      0.000       0.008       0.010
===================================================================================
Ljung-Box (L1) (Q):                   2.01   Jarque-Bera (JB):                59.39
Prob(Q):                              0.16   Prob(JB):                         0.00
Heteroskedasticity (H):               0.33   Skew:                            -0.17
Prob(H) (two-sided):                  0.00   Kurtosis:                         4.87
===================================================================================
```

잔차가 백색잡음 과정인지
정규성 및 등분산성을 만족하는지 파악

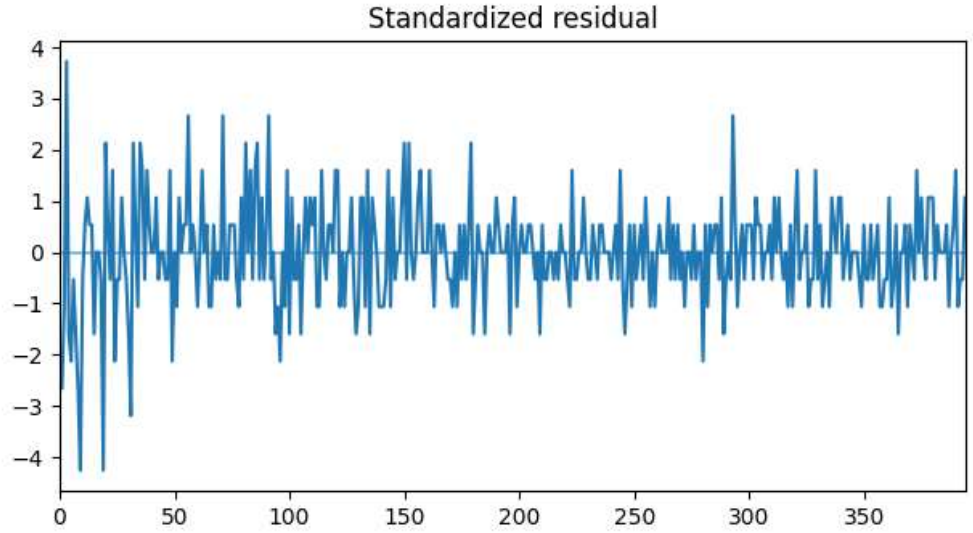
•Heteroskedasticity (H) 이분산성 검정 통계량
잔차가 이분산을 띠지 않는지 검정한 통계량
→ 회귀계수의 표준오차(분산)이 다르지 않다

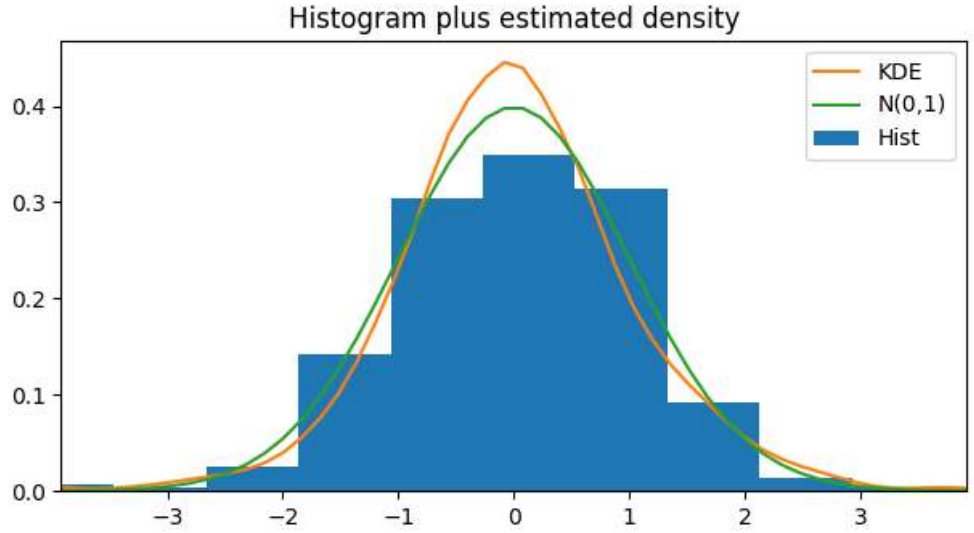
잔차가 정규분포를 따른다면, 경험적으로

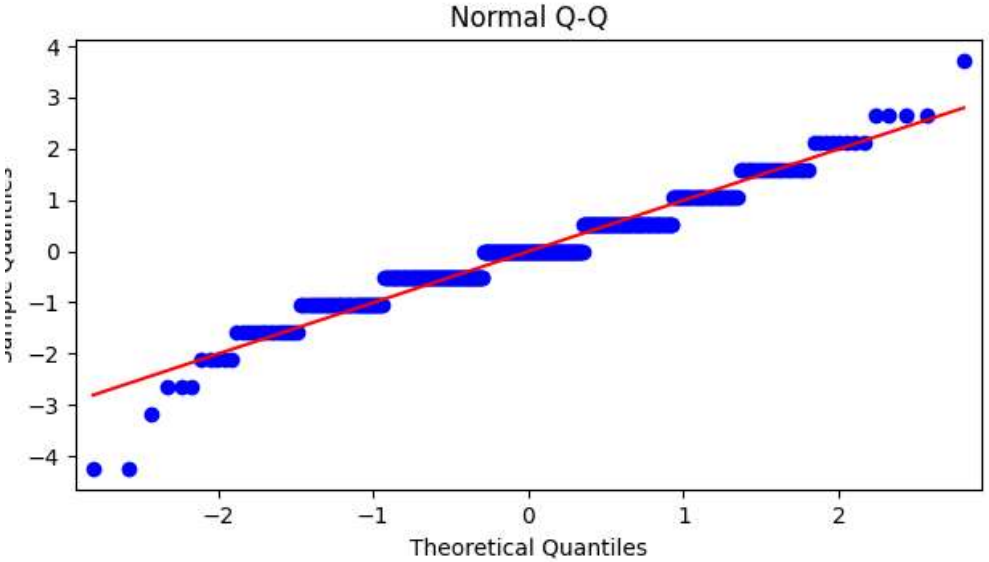
•비대칭도 (Skew)는 0에 가까워야
•첨도 (Kurtosis)는 3에 가까워야

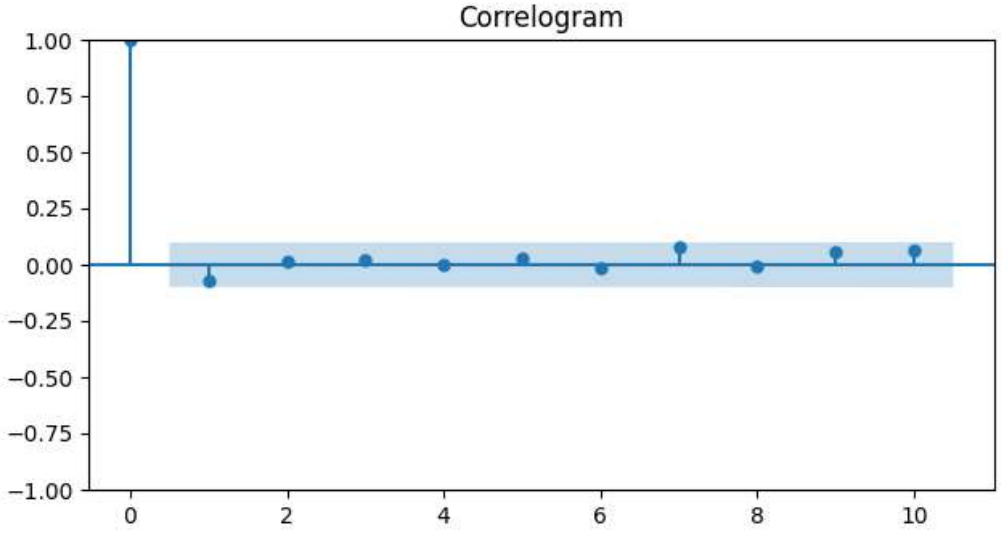
비대칭도는 -0.17으로 0에 가깝지만
첨도는 4.87로 3보다 더 높은 값을 가지고 있다

→ 잔차 비 정규 분포



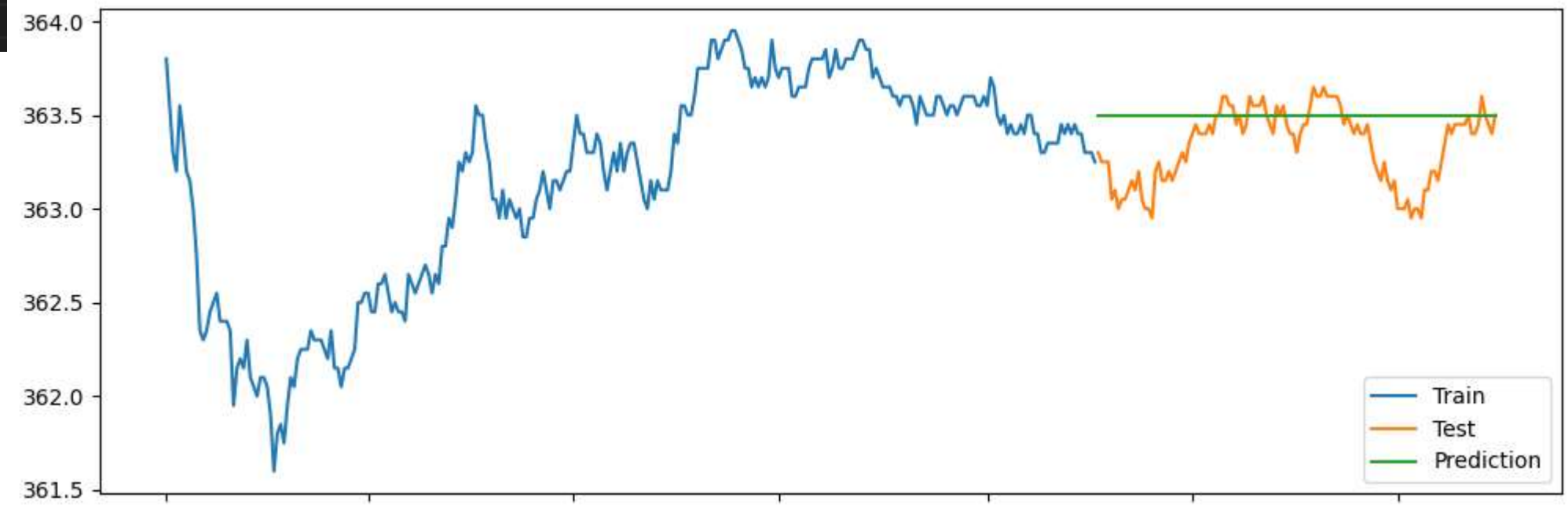






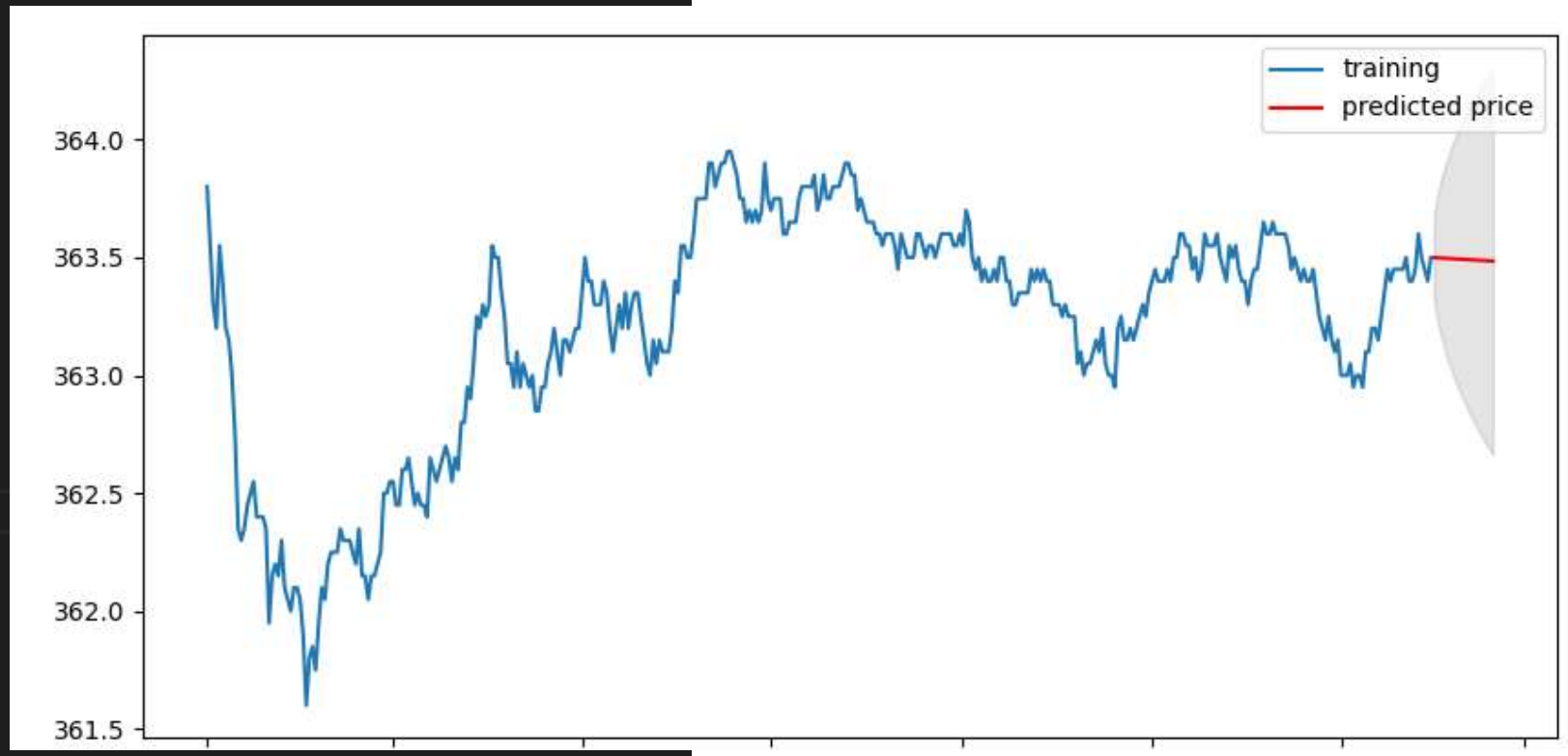
적합한 ARIMA (0,1,0)으로 남은 잔차는 백색 잡음이지만, 정규성은 따르지 않는다

```python
# 테스트 데이터 개수만큼 예측
train_data, test_data = ts_log[:int(len(ts_log)*0.7)], ts_log[int(len(ts_log)*0.7):]

y_predict = best_model.predict(n_periods=len(test_data))
y_predict = pd.DataFrame(y_predict,index = test_data.index,columns=['Prediction'])

# 그래프
fig, axes = plt.subplots(1, 1, figsize=(12, 4))
plt.plot(train_data, label='Train')        # 훈련 데이터
plt.plot(test_data, label='Test')          # 테스트 데이터
plt.plot(y_predict, label='Prediction')  # 예측 데이터
plt.legend()
plt.show()
```



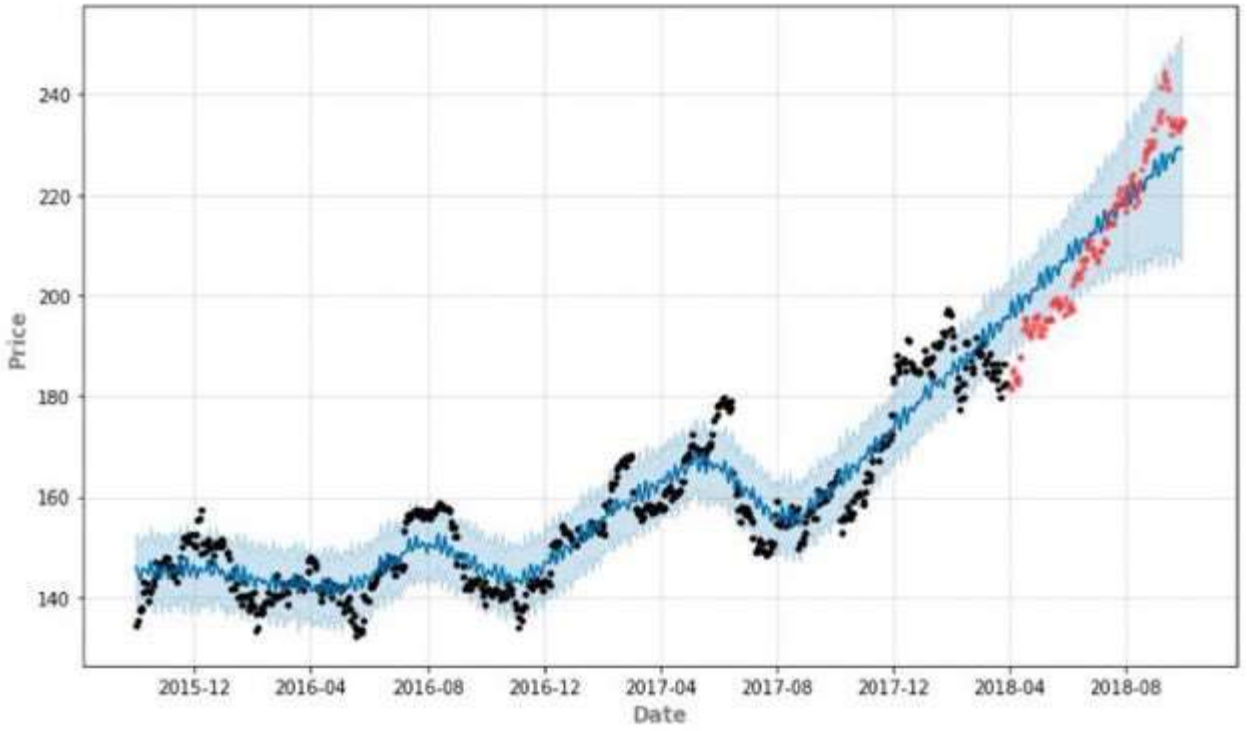
```python
from statsmodels.tsa.arima_model import ARIMA
model = ARIMA(ts_log, order=(0, 1, 0))
fitted_m = model.fit(disp=-1)
predict_num=20

fc, se, conf = fitted_m.forecast(predict_num, alpha=0.05)  # 95% conf
fc_series = pd.Series(fc, index=dates)   # 예측결과
lower_series = pd.Series(conf[:, 0], index=dates)  # 예측결과의 하한 바운드
upper_series = pd.Series(conf[:, 1], index=dates)  # 예측결과의 상한 바운드

predict_price = round(fc_series[-1],2)
now_price = ts_log["Close"][-1]
range = round(abs(now_price - predict_price),2)
print("최종예측값")
print(predict_price)
print("현재가")
print(now_price)

if now_price > predict_price:
    print(range, " 포인트 하락 예상")
else:
    print(range, " 포인트 상승 예상")


plt.figure(figsize=(10,5), dpi=100)
plt.plot(ts_log, label='training')
plt.plot(fc_series, c='r',label='predicted price')
plt.fill_between(lower_series.index, lower_series, upper_series, color='k', alpha=.10)
plt.legend()
plt.show()
```

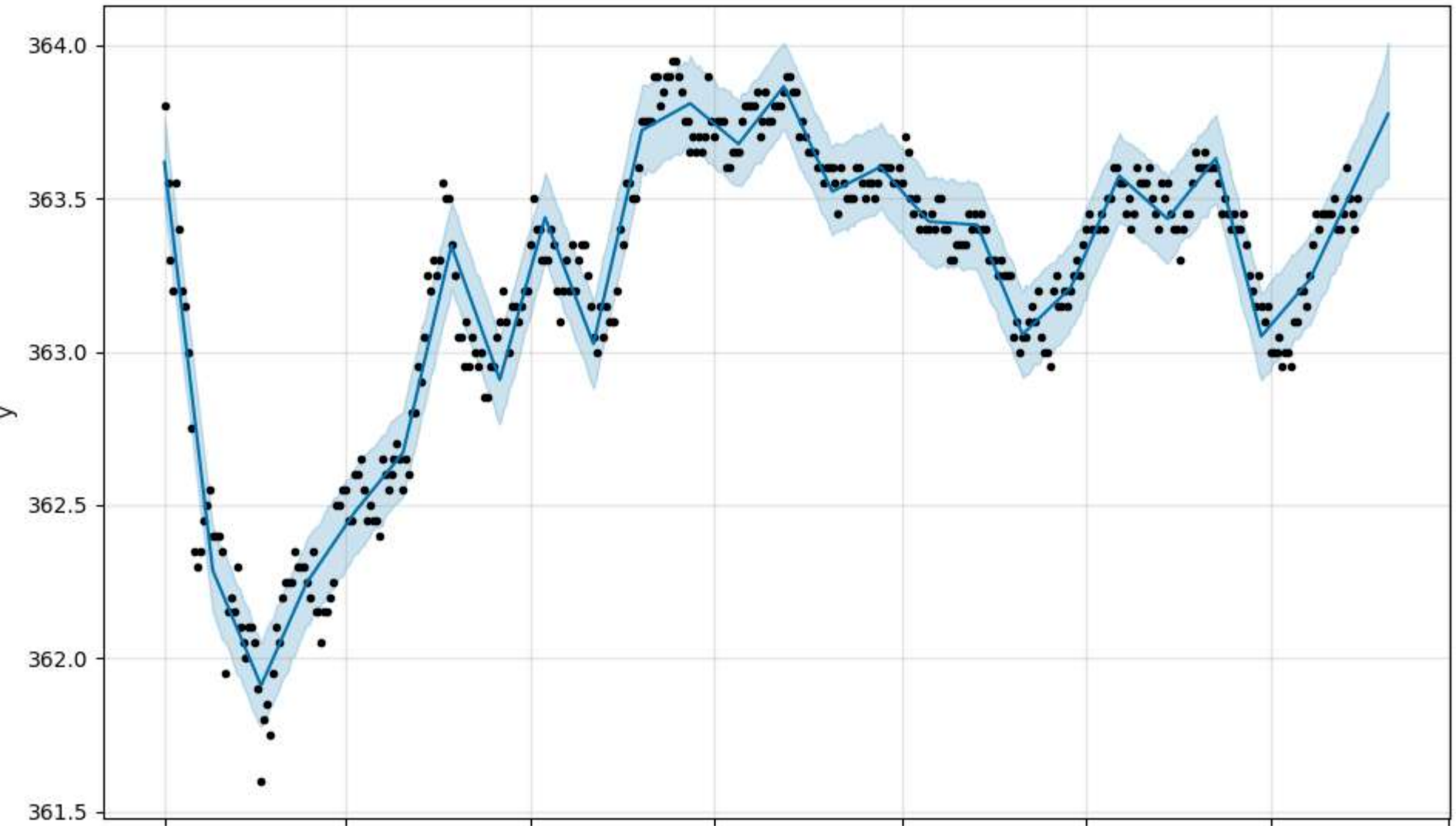
```
최종예측값
363.48
현재가
363.5
0.02  포인트 하락 예상
```



fbProphet


PROPHET


240
220
200
180
160
140
Price
2015-12
2016-04
2016-08
2016-12
2017-04
2017-08
2017-12
2018-04
2018-08
Date

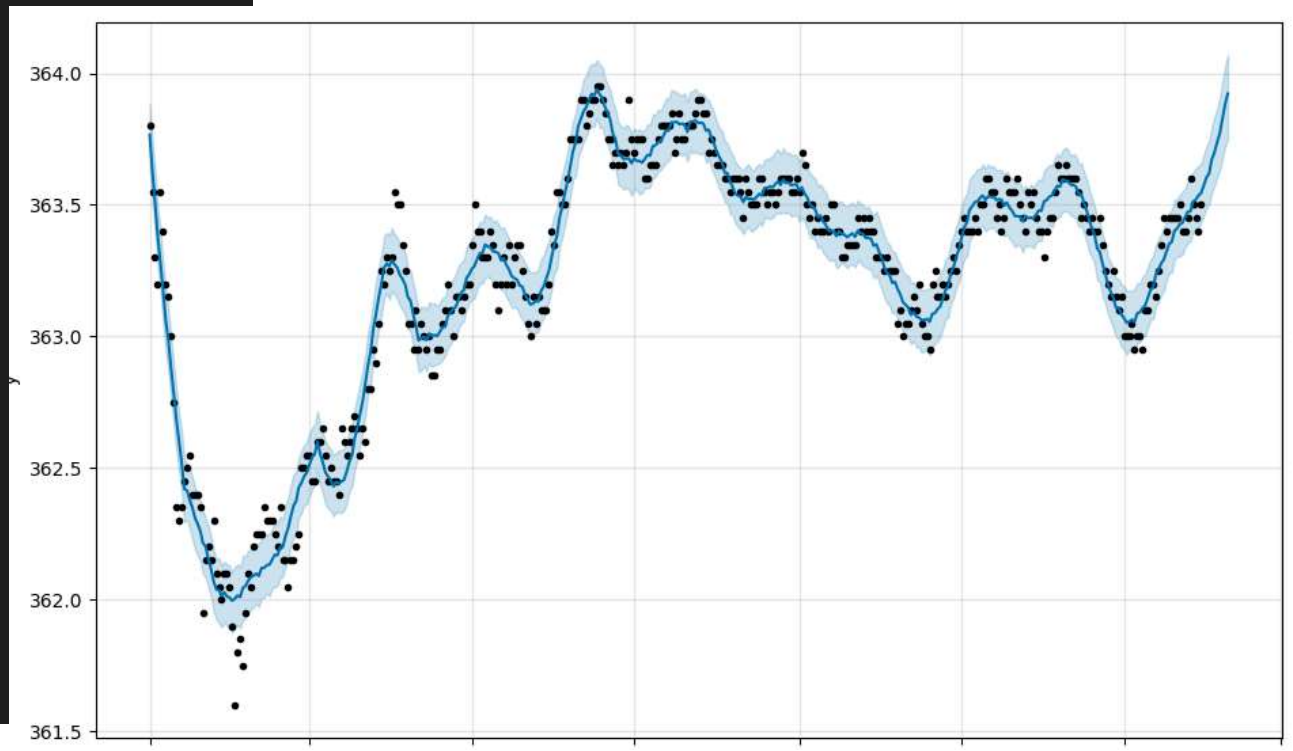


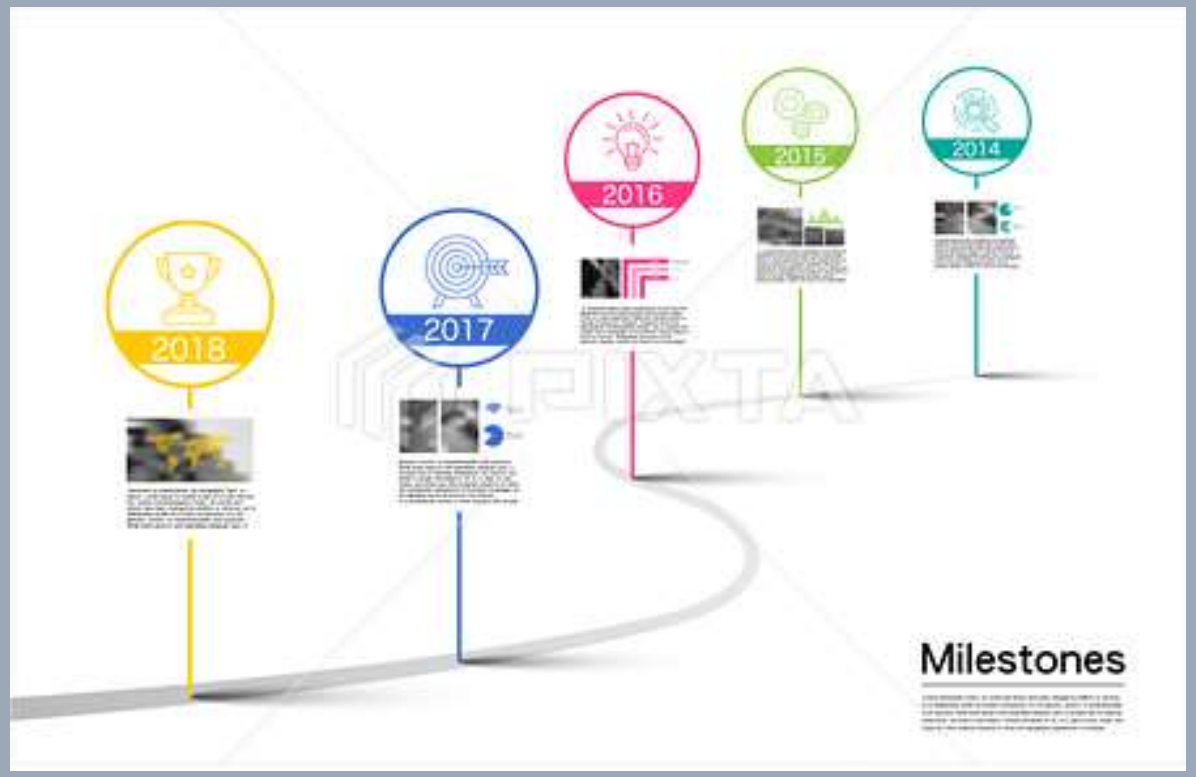

```
최종예측값
363.78
현재가
363.5
0.28  포인트 상승 예상
```

```
phet = Prophet(yearly_seasonality=False, daily_seasonality=True,weekly_seasonality=False, changepoint_range=1,changepoint_prior_scale=0.75)
```

```python
ts_log2 = ts_log.reset_index()
ts_log2.columns = ['ds', 'y']

prophet = Prophet(yearly_seasonality=True, daily_seasonality=True)
prophet.fit(ts_log2)

# 이후 periods 간의 데이터 예측
future = prophet.make_future_dataframe(periods=10)
forecast = prophet.predict(future)
#print(forecast[["ds", "yhat", "yhat_lower", "yhat_upper"]].tail())

predict_price = round(forecast[["yhat"]].iloc[-1],2).values[0]
now_price = ts_log["Close"][-1]
range = round(abs(now_price - predict_price),2)

print("최종예측값")
print(predict_price)
print("현재가")
print(now_price)

if now_price > predict_price:
    print(range, " 포인트 하락 예상")
else:
    print(range, " 포인트 상승 예상")

prophet.plot(forecast)
plt.show()
```

```
최종예측값
363.92
현재가
363.5
0.42  포인트 상승 예상
```





Content

0 | 팀원 소개

1 | 프로젝트 진행 상황 리뷰

2 | 테크니컬 리뷰

3 | 진행 예정 프로세스 소개

4 | 시연 예시 / Q & A

Content

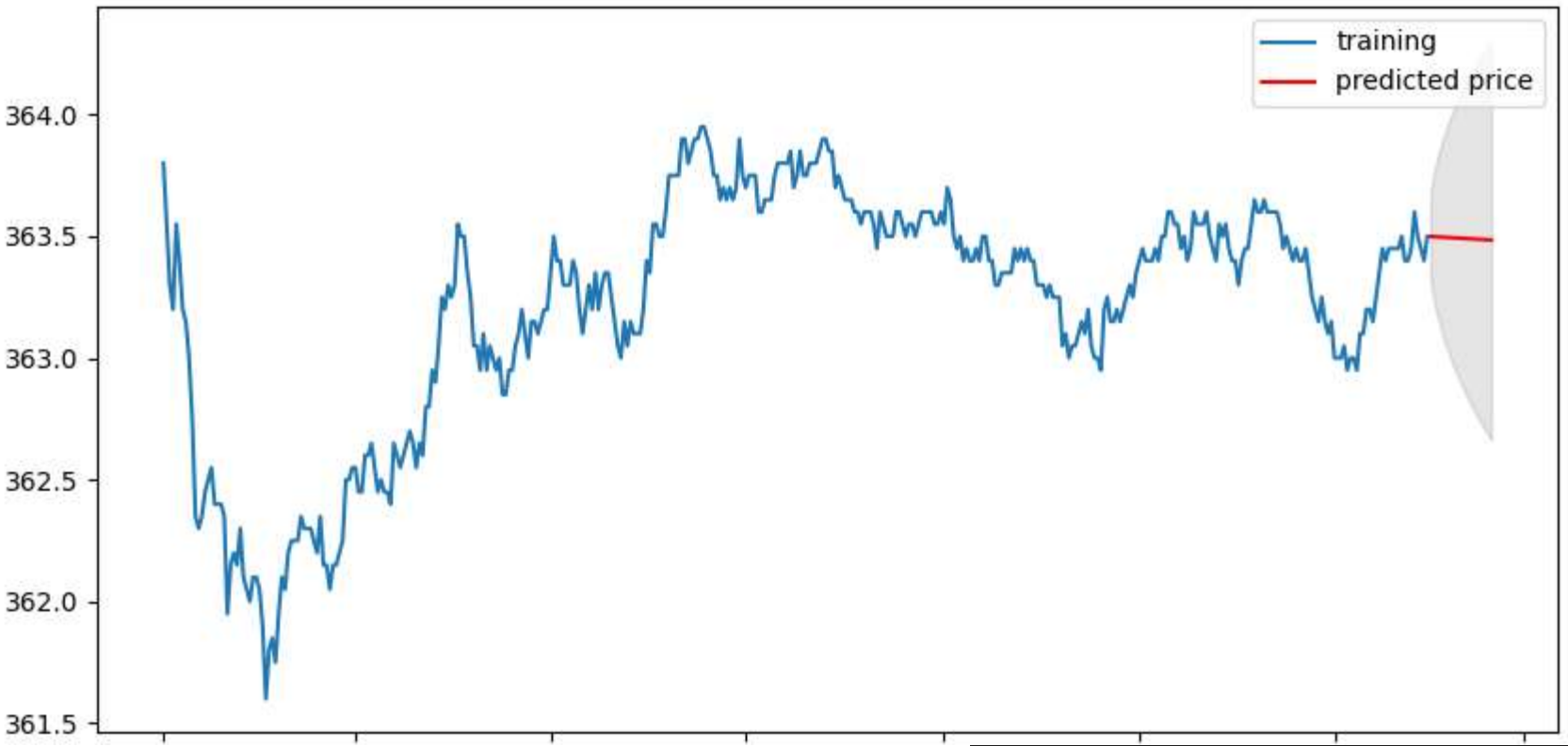
0 | 팀원 소개

1 | 프로젝트 진행 상황 리뷰

2 | 테크니컬 리뷰

3 | 진행 예정 프로세스 소개

4 | 시연 예시 / Q & A



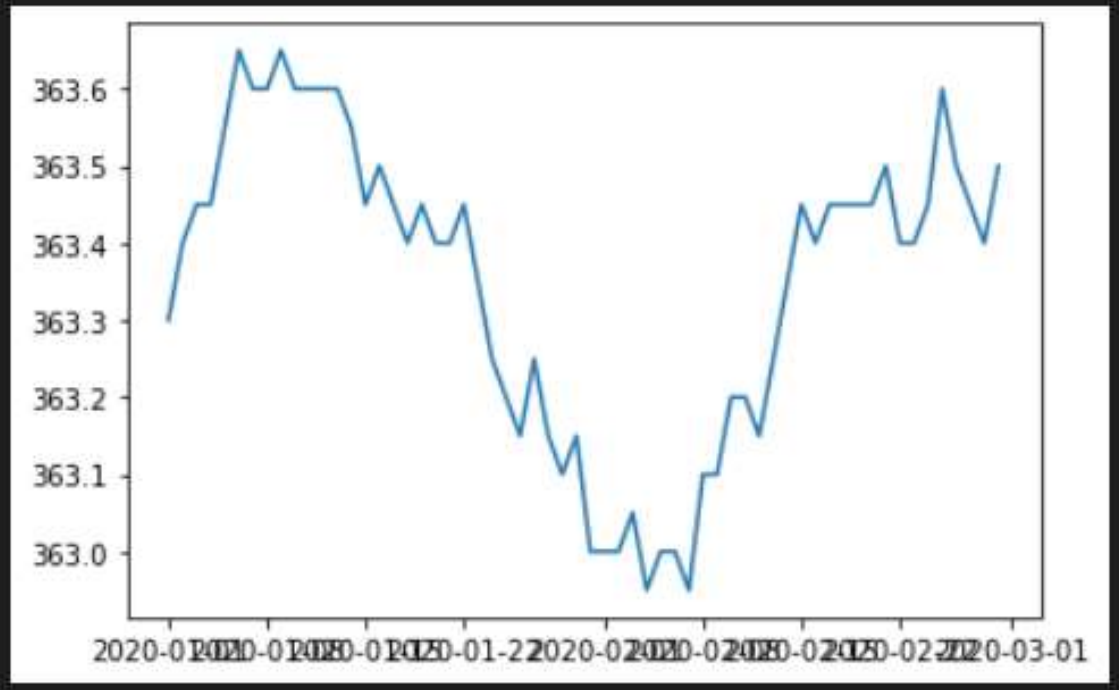
최종예측값
363.48
현재가
363.5
0.02 포인트 하락 예상

```
STEP 1 : 시계열 데이터 준비
Kospi200 1분봉 종가 60개
```

```
train_data, test_data = ts_log[:int(len(ts_log)*0.9)], ts_log[int(len(ts_log)*0.9):]
plt.plot(ts_log)
```

[2] ✓ 0.3s

```
[<matplotlib.lines.Line2D at 0x1af36fd5340>]
```



```
STEP 2 : 시계열 안정성 분석
정성적 그래프 분석
정량적 Augmented Dicky-Fuller Test
시계열 분해(Time Series Decomposition)
Residual 안정성 확인
```

```python
# 일정 시간 내 구간 통계치(Rolling Statistics)를 시각화해 보는 함수
def plot_rolling_statistics(timeseries, window=12):

    rolmean = timeseries.rolling(window=window).mean()  # 이동평균 시계열
    rolstd = timeseries.rolling(window=window).std()    # 이동표준편차 시계열

    # 원본시계열, 이동평균, 이동표준편차를 plot으로 시각화해 본다.
    orig = plt.plot(timeseries, color='blue',label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    std = plt.plot(rolstd, color='black', label = 'Rolling Std')

    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)
```

[4] ✓ 0.4s

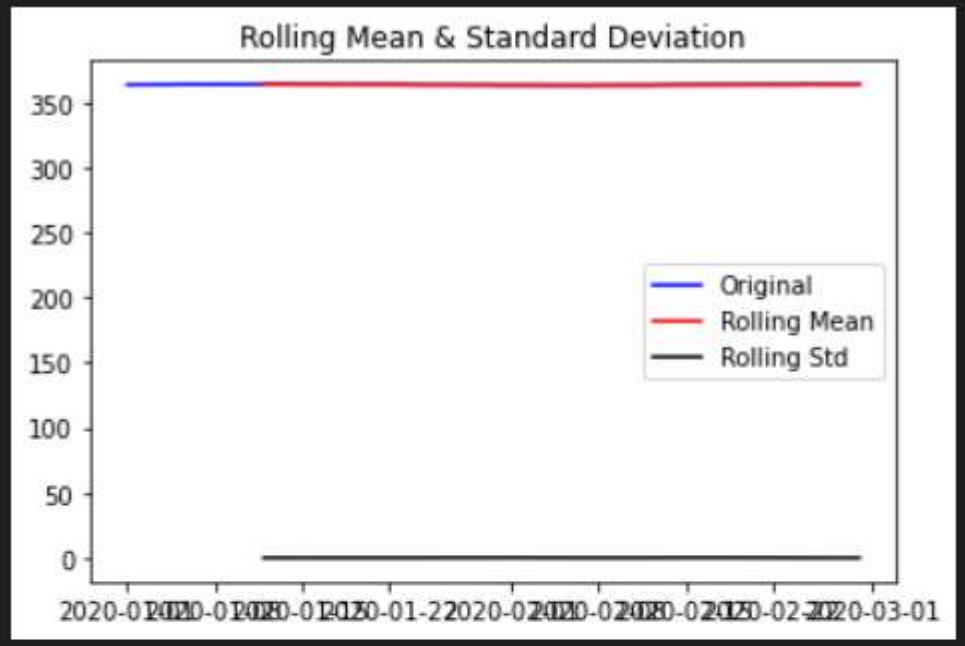
```python
# 주어진 timeseries에 대한 Augmented Dickey-Fuller Test를 수행하는 함수
from statsmodels.tsa.stattools import adfuller

def augmented_dickey_fuller_test(timeseries):
    # statsmodels 패키지에서 제공하는 adfuller 메소드를 호출합니다.
    dftest = adfuller(timeseries, autolag='AIC')

    # adfuller 메소드가 리턴한 결과를 정리하여 출력합니다.
    print('Results of Dickey-Fuller Test:')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)' % key] = value
    print(dfoutput)
```

✓ 0.7s

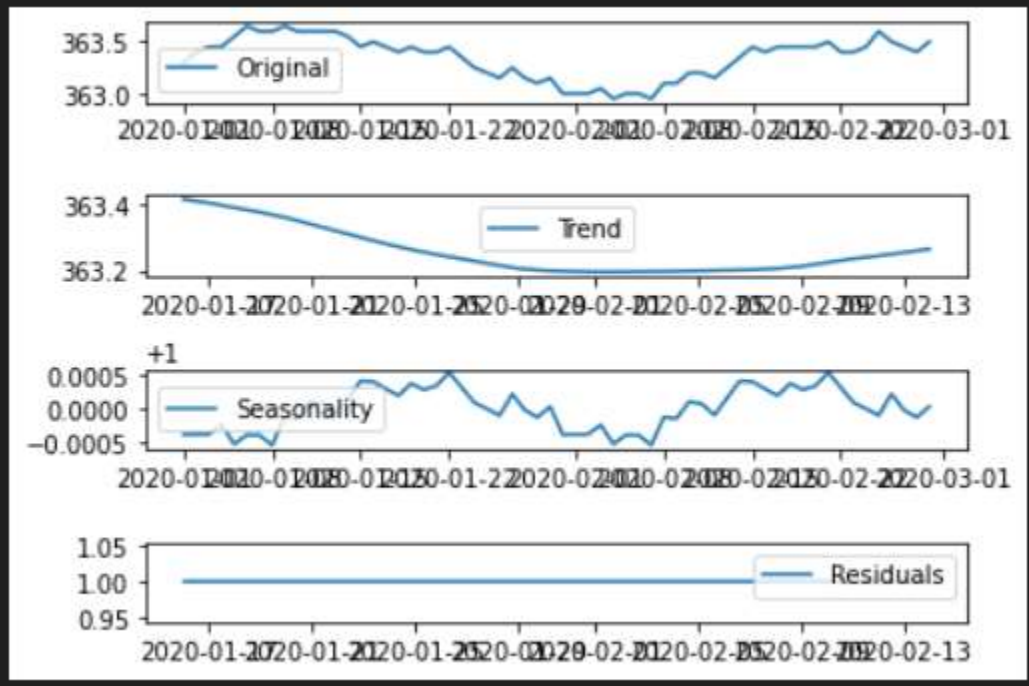
```
print()
print('\033[31m'+'\033[1m' + "정성적 그래프 분석 :" + '\033[0m')
# 정성적 그래프 분석
plot_rolling_statistics(ts_log, window=12)

print()
print('\033[31m'+'\033[1m' + "정량적 Augmented Dicky-Fuller Test :" + '\033[0m')
#정량적 Augmented Dicky-Fuller Test
augmented_dickey_fuller_test(ts_log)

print()
print('\033[31m'+'\033[1m' + "시계열 분해 (Time Series Decomposition) :" + '\033[0m')
#시계열 분해 (Time Series Decomposition)
from statsmodels.tsa.seasonal import seasonal_decompose

decomposition = seasonal_decompose(ts_log, model='multiplicative', period = 30)

trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid

plt.subplot(411)
plt.plot(ts_log, label='Original')
plt.legend(loc='best')
plt.subplot(412)
plt.plot(trend, label='Trend')
plt.legend(loc='best')
plt.subplot(413)
plt.plot(seasonal,label='Seasonality')
plt.legend(loc='best')
plt.subplot(414)
plt.plot(residual, label='Residuals')
plt.legend(loc='best')
plt.tight_layout()
```

✓ 0.8s

정성적 그래프 분석 :



시계열 분해 (Time Series Decomposition) :



정량적 Augmented Dicky-Fuller Test :

Results of Dickey-Fuller Test:

Test Statistic	-1.234295
p-value	0.658672
#Lags Used	0.000000
Number of Observations Used	59.000000
Critical Value (1%)	-3.546395
Critical Value (5%)	-2.911939
Critical Value (10%)	-2.593652

dtype: float64

Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting

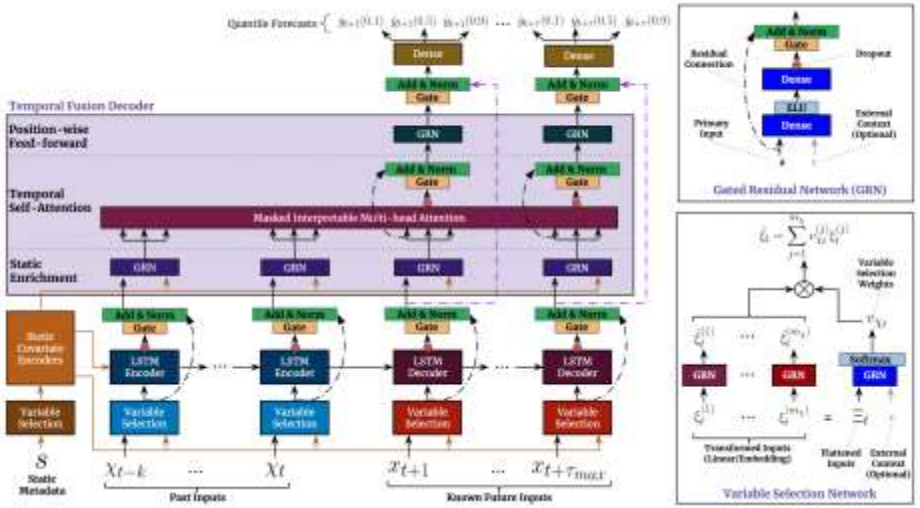
Bryan Lim[a,1,*], Sercan Ö. Arık[b], Nicolas Loeff[b], Tomas Pfister[b]

[a] *University of Oxford, UK*
[b] *Google Cloud AI, USA*

Abstract

Multi-horizon forecasting often contains a complex mix of inputs – including static (i.e. time-invariant) covariates, known future inputs, and other exogenous time series that are only observed in the past – without any prior information on how they interact with the target. Several deep learning methods have been proposed, but they are typically 'black-box' models which do not shed light on how they use the full range of inputs present in practical scenarios. In this paper, we introduce the Temporal Fusion Transformer (TFT) – a novel attention-based architecture which combines high-performance multi-horizon forecasting with interpretable insights into temporal dynamics. To learn temporal relationships at different scales, TFT uses recurrent layers for local processing and interpretable self-attention layers for long-term dependencies. TFT utilizes specialized components to select relevant features and a series of gating layers to suppress unnecessary components, enabling high performance in a wide range of scenarios. On a variety of real-world datasets, we demonstrate significant performance improvements over existing benchmarks, and showcase three practical interpretability use cases of TFT.

Keywords: Deep learning, Interpretability, Time series, Multi-horizon forecasting, Attention mechanisms, Explainable AI.



2021 춘계학술발표대회 논문집 제28권 제1호 (2021. 5)

Temporal Fusion Transformer 모델을 활용한 다층 수평 시계열 데이터 분석

김인경, 김대희, 이재구*
국민대학교 컴퓨터공학과
*jaekoo@kookmin.ac.kr

Multi-horizon Time Series Forecasting Using Temporal Fusion Transformer

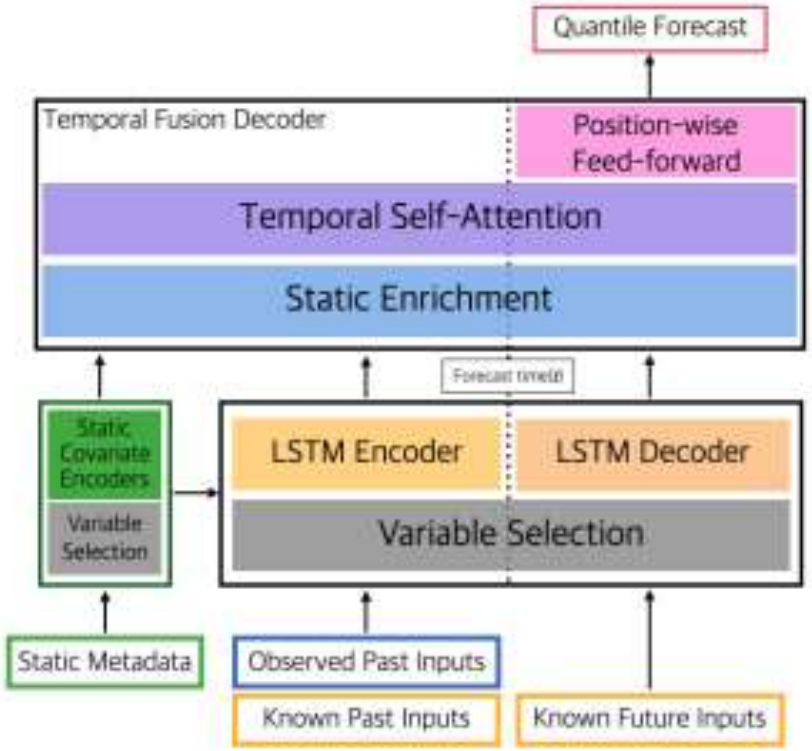
Inkyung Kim, Daehee Kim, Jaekoo Lee*
Dep. of Computer Science, Kookmin University

요 약

시계열 형태의 데이터는 다양한 분야에서 수집되고 응용되기 때문에 정확한 시계열 예측은 많은 분야에서 운영 효율성을 높일 수 있는 중요한 분석 방법으로 고려된다. 그중 다층 수평 예측은 사용자에게 전반적인 시계열 데이터 경향성을 제공할 수 있다. 하지만 다양한 정보를 포함하는 시계열 데이터는 데이터에 내재한 이질성(heterogeneity)까지 포괄적으로 고려한 방법을 통해서만 정확한 예측을 할 수 있다. 하지만 지금까지 많은 시계열 분석 모델들이 데이터의 이질성을 반영하지 못했다. 이러한 한계를 보완하고자 우리는 Temporal Fusion Transformer 모델을 사용하여 실생활과 밀접한 관련이 있는 데이터에 적용하여 이질성을 고려한 향상된 예측을 수행하였다. 실제, 주식 데이터와 미세 먼지 데이터와 같은 실생활 시계열 데이터에 적용하였고 실험 결과 기존 모델보다 Mean Squared Error(MSE)가 0.3487 낮은 것을 확인하였다.

3. TFT 를 활용한 시계열 데이터 예측

3.1 Temporal Fusion Transformer(TFT) 모델



(그림 1) TFT 모델[1] 구조.

```
trainer.fit(tft)
```

[67] 955m 23.3s

```
GPU available: True, used: False
TPU available: False, using: 0 TPU cores
```

	Name	Type	Params
0	categorical_var_embeddings	ModuleList	59.0 K
1	regular_var_embeddings	ModuleList	1.3 K
2	static_context_variable_selection_grn	GatedResidualNetwork	103 K
3	static_context_enrichment_grn	GatedResidualNetwork	103 K
4	static_context_state_h_grn	GatedResidualNetwork	103 K
5	static_context_state_c_grn	GatedResidualNetwork	103 K
6	static_vsn	VariableSelectionNetwork	155 K
7	temporal_historical_vsn	VariableSelectionNetwork	571 K
8	temporal_future_vsn	VariableSelectionNetwork	440 K
9	historical_lstm	LSTM	206 K
10	future_lstm	LSTM	206 K
11	post_seq_encoder_gate_add_norm	GateAddNormNetwork	51.8 K
12	static_enrichment	GatedResidualNetwork	128 K
13	self_attn_layer	InterpretableMultiHeadAttention	64.0 K
14	post_attn_gate_add_norm	GateAddNormNetwork	51.8 K
15	GRN_positionwise	GatedResidualNetwork	103 K
16	post_tfd_gate_add_norm	GateAddNormNetwork	51.8 K
17	output_feed_forward	Linear	483

```
2.5 M     Trainable params
0         Non-trainable params
2.5 M     Total params
10.023    Total estimated model params size (MB)
```