

## 비트코인 가격 분석

## #01. 작업준비

## 패키지 참조

## #02 pyupbit 패키지 사용

## 조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

## 비트코인 가격 분석

## #01. 작업준비

## 패키지 참조

pyupbit 패키지의 설치가 필요하다.

```
import sys
sys.path.append("../..")

import pyupbit
import datetime as dt
from matplotlib import pyplot as plt
from matplotlib import dates as mdates
from statsmodels.tsa.arima.model import ARIMA
from pmdarima.arima import auto_arima
import seaborn as sb

from helper import set_datetime_index, exp_time_data
```

## #02 pyupbit 패키지 사용

## 조회 가능한 단위 목록

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

- KRW : 한국화폐
- BTC : Bitcoin
- ETH : 이더리움

```
print(pyupbit.get_tickers())
```

```
['KRW-BTC', 'KRW-ETH', 'BTC-ETH', 'BTC-XRP', 'BTC-ETC', 'BTC-CVC', 'BTC-
```

## 특정 단어를 포함하는 조회 가능 목록만 가져오기

```
print(pyupbit.get_tickers(fiat='KRW'))
```

```
['KRW-BTC', 'KRW-ETH', 'KRW-NEO', 'KRW-MTL', 'KRW-XRP', 'KRW-ETC', 'KRW-
```

## 현재 시세 가져오기

조회하고자 하는 단위의 이름을 리스트로 설정한다.

```
pyupbit.get_current_price(["KRW-BTC", "KRW-ETH"])
```

```
{'KRW-BTC': 38545000.0, 'KRW-ETH': 2435000.0}
```

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

## 특정 기간에 대한 시세 데이터 가져오기

비트코인의 한국 화폐단위( KRW-BTC ) 데이터를 오늘 날짜부터 500일간 일단위( day )로 조회

조회간격 단위: minute1, minute3, minute5, minute10, minute15, minute30, minute60, minute240, day, week, month

필드	설명
open	시가
high	고가
low	저가
close	종가
volume	거래량
value	거래량을 지정된 화폐 단위로 환산한 가치

```

ticker = 'KRW-BTC'
to = dt.datetime.now().strftime('%Y-%m-%d')
count = 500
interval = 'day'
df = pyupbit.get_ohlcv(ticker=ticker, interval=interval, to=to, count=count)
df.head()

```

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검증

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

	open	high	low	close	volume	
2022-03-25 09:00:00	52907000.0	54450000.0	52700000.0	53791000.0	5670.827515	3.03815
2022-03-26 09:00:00	53836000.0	54241000.0	53402000.0	53902000.0	3333.058744	1.79588
2022-03-27 09:00:00	53888000.0	56201000.0	53674000.0	55949000.0	5341.023716	2.92027
2022-03-28 09:00:00	55936000.0	57678000.0	55827000.0	56914000.0	6374.435146	3.61568
2022-03-29 09:00:00	56900000.0	57540000.0	56390000.0	56985000.0	6629.178471	3.78907

## #03. 데이터 전처리

## 시세가격에 대한 파생변수 추가

최고가와 최저가의 평균을 그날의 시세가격으로 삼기로 결정하고 데이터 전처리를 수행

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

```
df['price'] = (df['high'] + df['low']) / 2
df.head()
```

	open	high	low	close	volume	
2022-03-25 09:00:00	52907000.0	54450000.0	52700000.0	53791000.0	5670.827515	3.03815
2022-03-26 09:00:00	53836000.0	54241000.0	53402000.0	53902000.0	3333.058744	1.79588
2022-03-27 09:00:00	53888000.0	56201000.0	53674000.0	55949000.0	5341.023716	2.92027
2022-03-28 09:00:00	55936000.0	57678000.0	55827000.0	56914000.0	6374.435146	3.61568
2022-03-29 09:00:00	56900000.0	57540000.0	56390000.0	56985000.0	6629.178471	3.78907

## 인덱스에 대한 날짜 형식 지정

```
set_datetime_index(df, inplace=True)
```

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

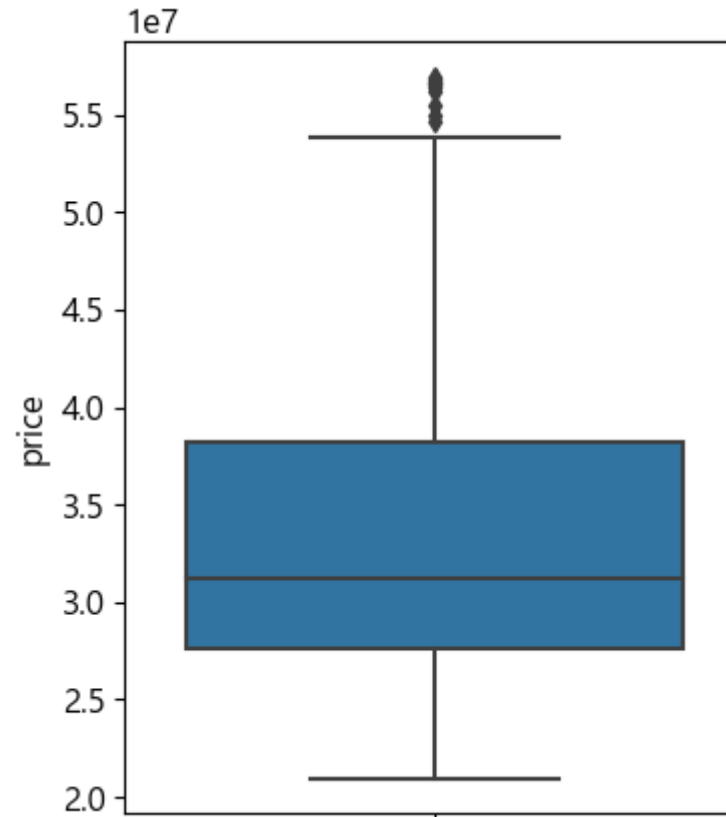
## #05. AutoARIMA 사용

## #04. 데이터 검정

이상치는 보이지만 데이터 자체가 실제 거래 데이터이므로 정상 데이터로 판단함.

```
exp_time_data(df, 'price', "m")
```

결측치 수: 0



## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

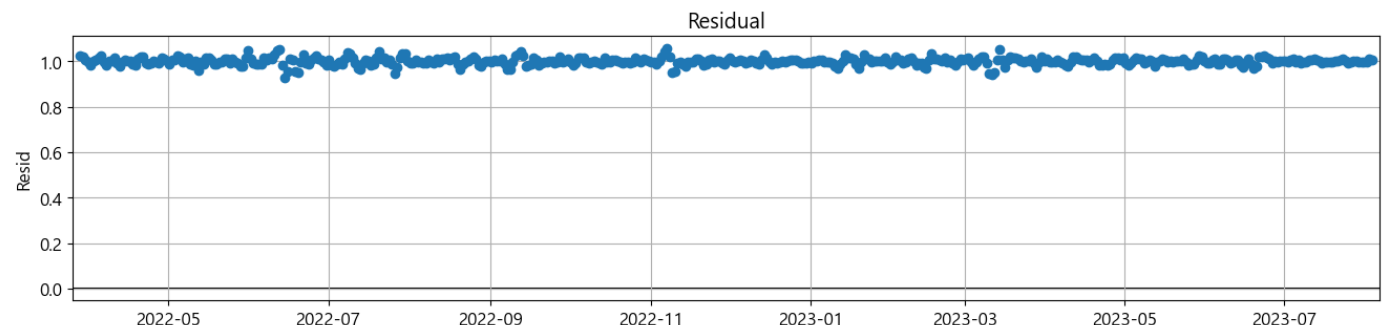
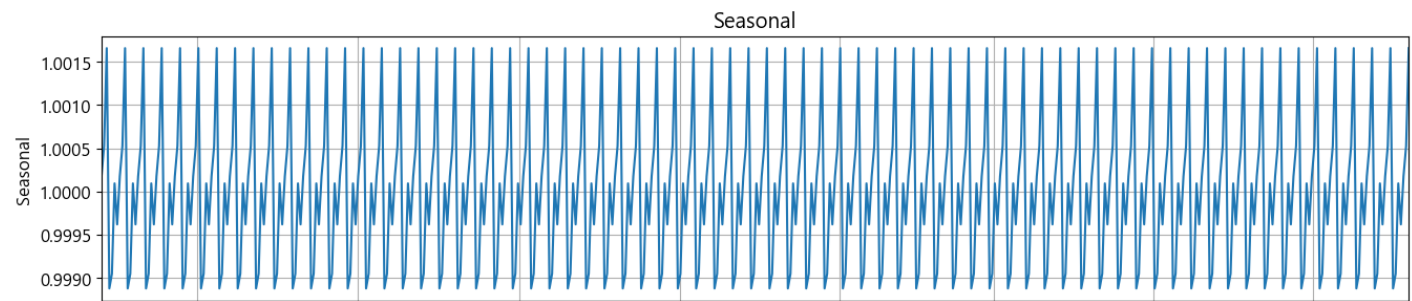
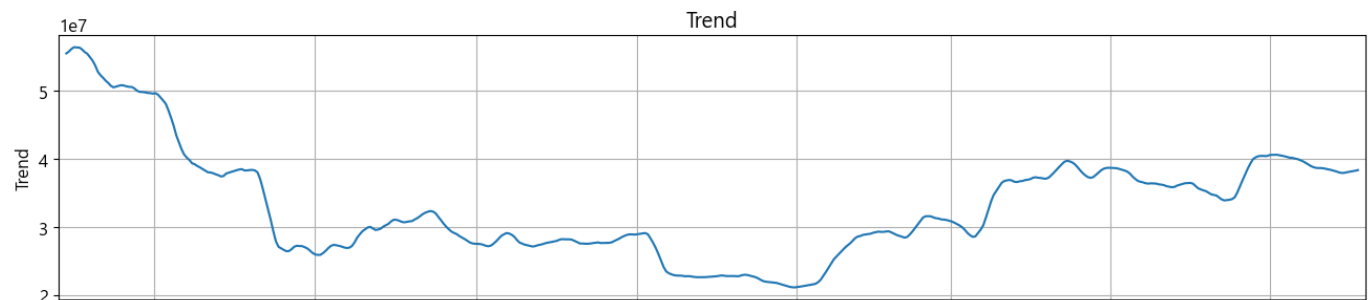
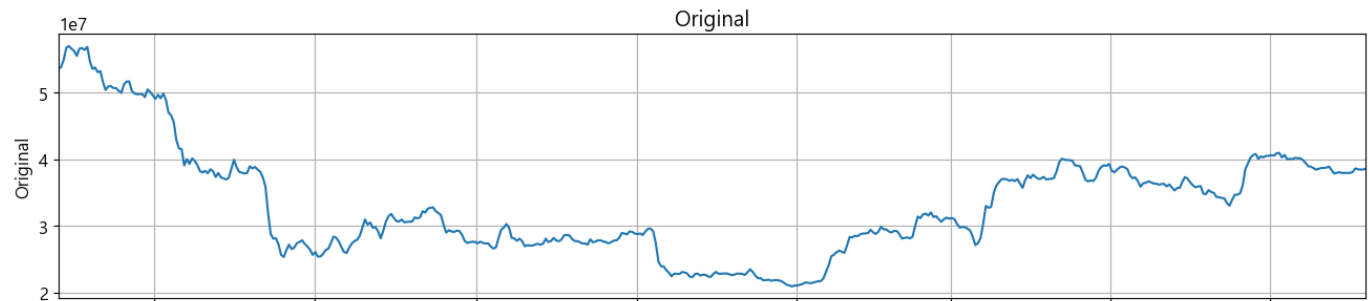
분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

07\_비트코인\_가격분석.ipynb



## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

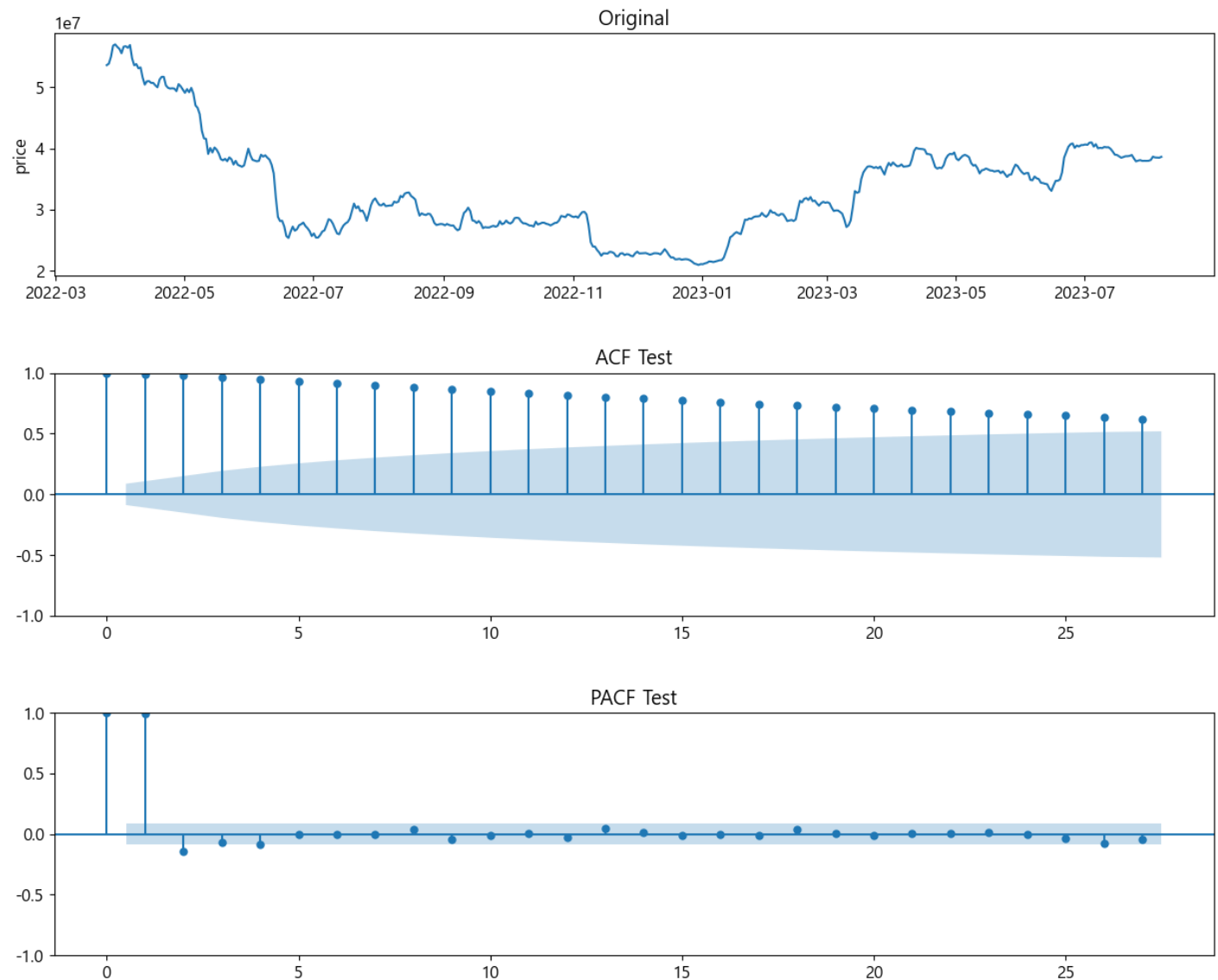
분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용



===== 원본 데이터 =====

```

+-----+-----+
| ADF Test |
+-----+-----+

```



## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

```
| 검정통계량(ADF Statistic) | -3.07508 |
| 유의수준(p-value) | 0.0284529 |
| 최적차수(num of lags) | 3 |
| 관측치 개수(num of observations) | 496 |
| 기각값(Critical Values) 1% | -3.4436 |
| 기각값(Critical Values) 5% | -2.86738 |
| 기각값(Critical Values) 10% | -2.56988 |
| 데이터 정상성 여부(0=False,1=True) | 1 |
+-----+-----+
```

## #04. ARIMA 분석

## 분석 모델 구축

```
model = ARIMA(df['price'], order=(1,0,0), seasonal_order=(1,1,0,7))
fit = model.fit()
print(fit.summary())
```

## SARIMAX Results

```
Dep. Variable: price No. Observations:
Model: ARIMA(1, 0, 0)x(1, 1, 0, 7) Log Likelihood
Date: Mon, 07 Aug 2023 AIC
Time: 11:51:57 BIC
Sample: 03-25-2022 HQIC
- 08-06-2023

Covariance Type: opg
```

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	0.6069	0.002	359.813	0.000	0.604	0.609
ar.S.L7	-0.2590	0.002	-112.625	0.000	-0.263	-0.255
sigma2	8.931e+11	5.46e-16	1.64e+27	0.000	8.93e+11	8.93e+11
Ljung-Box (L1) (Q):						
			68.82	Jarque-Bera (JB):		
Prob(Q):			0.00	Prob(JB):		
Heteroskedasticity (H):			0.17	Skew:		
Prob(H) (two-sided):			0.00	Kurtosis:		

## Warnings:

- [1] Covariance matrix calculated using the outer product of gradients (cond number: 1.1e+16)
- [2] Covariance matrix is singular or near-singular, with condition number of 1.1e+16. This may be caused by perfect collinearity in the design matrix X. Please check for collinearity in the design matrix X. This may be caused by perfect collinearity in the design matrix X.

## 분석 모델에 대한 예측치

```
fittedvalues = fit.fittedvalues
fittedvalues.head()
```

```
2022-03-25 09:00:00    0.000000e+00
2022-03-26 09:00:00    3.207295e+07
2022-03-27 09:00:00    3.193289e+07
2022-03-28 09:00:00    3.212577e+07
```

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

```
2022-03-29 09:00:00    3.240832e+07
Freq: D, dtype: float64
```

## 향후 30일간의 데이터 예측

```
fc = fit.forecast(30)
fc.head()
```

```
2023-08-07 09:00:00    3.831736e+07
2023-08-08 09:00:00    3.818717e+07
2023-08-09 09:00:00    3.856191e+07
2023-08-10 09:00:00    3.844897e+07
2023-08-11 09:00:00    3.837836e+07
Freq: D, Name: predicted_mean, dtype: float64
```

## 시각화

```
plt.figure(figsize=(20,8))

sb.lineplot(data=df, x=df.index, y='price', label='Original')
sb.lineplot(x=fv.index, y=fv.values, label='FittedValues', linestyle='--')
sb.lineplot(x=fc.index, y=fc.values, label='Predict', linestyle='--', color='red')

plt.xlabel('Month')
plt.ylabel('Price')
plt.legend()
```

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

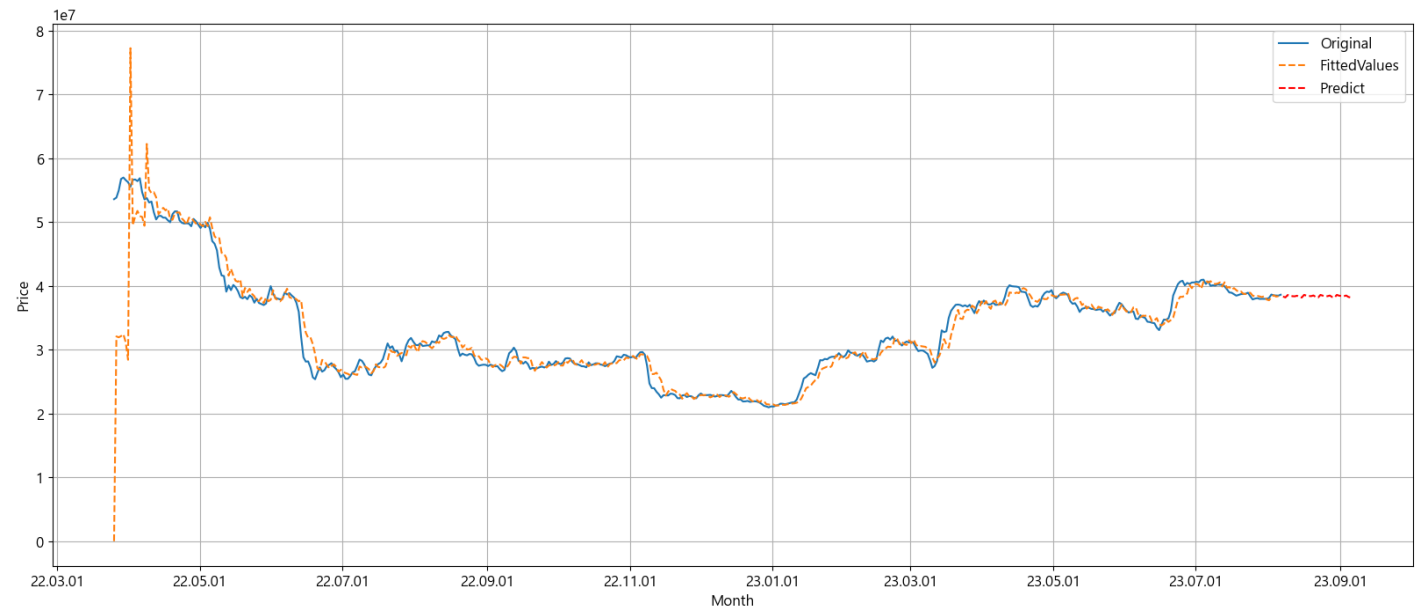
향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

```
monthyearFmt = mdates.DateFormatter('%y.%m.%d')
plt.gca().xaxis.set_major_formatter(monthyearFmt)
```

```
plt.grid()
plt.show()
plt.close()
```



## #05. AutoARIMA 사용

## 분석모델 구축

```
my_p = 1 # AR의 차수 (검증한 결과를 활용)
my_d = 1 # 차분 횟수 (검증한 결과를 활용)
my_q = 0 # MA의 차수 (검증한 결과를 활용)
```

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검증

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

```
my_s = 7 # 계절성 주기 (분석가가 판단)
```

```
model = auto_arima(
    y=df['price'], # 모델링하려는 시계열 데이터 또는 배열
    start_p=0, # p의 시작점
    max_p=my_p, # p의 최대값
    d=my_d, # 차분 횟수
    start_q=0, # q의 시작점
    max_q=my_q, # q의 최대값
    seasonal=True, # 계절성 사용 여부
    m=my_s, # 계절성 주기
    start_P=0, # P의 시작점
    max_P=my_p, # P의 최대값
    D=my_d, # 계절성 차분 횟수
    start_Q=0, # Q의 시작점
    max_Q=my_q, # Q의 최대값
    trace=True # 학습 과정 표시 여부
)
print(model.summary())
```

Performing stepwise search to minimize aic

```
ARIMA(0,1,0)(0,1,0)[7] : AIC=14956.004, Time=0.02 sec
ARIMA(1,1,0)(1,1,0)[7] : AIC=14929.788, Time=0.12 sec
ARIMA(1,1,0)(0,1,0)[7] : AIC=14957.743, Time=0.04 sec
ARIMA(0,1,0)(1,1,0)[7] : AIC=14949.560, Time=0.16 sec
ARIMA(1,1,0)(1,1,0)[7] intercept : AIC=14931.767, Time=0.16 sec
```

Best model: ARIMA(1,1,0)(1,1,0)[7]

Total fit time: 0.506 seconds

SARIMAX Results

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검증

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

```

Dep. Variable:          y      No. Observations:
Model:                SARIMAX(1, 1, 0)x(1, 1, 0, 7)  Log Likelihood
Date:                  Mon, 07 Aug 2023             AIC
Time:                  11:52:07                     BIC
Sample:                03-25-2022                   HQIC
                    - 08-06-2023

Covariance Type:                opg

```

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	0.0420	0.005	7.769	0.000	0.031	0.053
ar.S.L7	-0.0844	0.004	-21.001	0.000	-0.092	-0.077
sigma2	8.058e+11	1.15e-16	7.03e+27	0.000	8.06e+11	8.06e+11

```

Ljung-Box (L1) (Q):                54.67      Jarque-Bera (JB):
Prob(Q):                          0.00      Prob(JB):
Heteroskedasticity (H):            0.43      Skew:
Prob(H) (two-sided):              0.00      Kurtosis:

```

## Warnings:

```

[1] Covariance matrix calculated using the outer product of gradients (cond=1e+27)
[2] Covariance matrix is singular or near-singular, with condition number=1e+27

```

## 잔차 분석

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

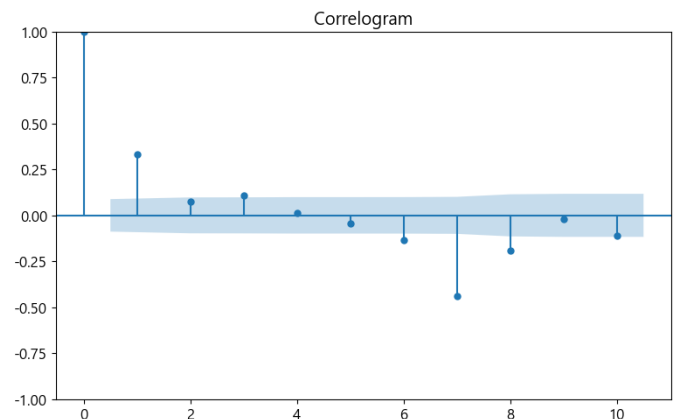
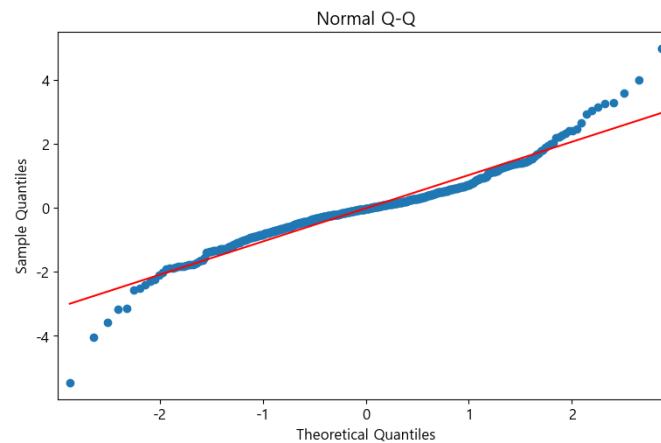
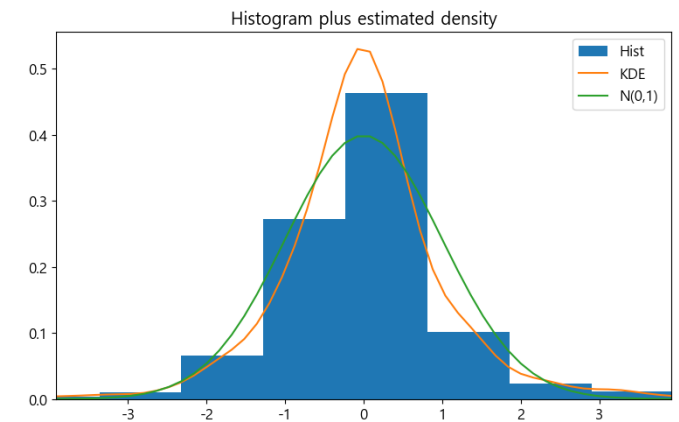
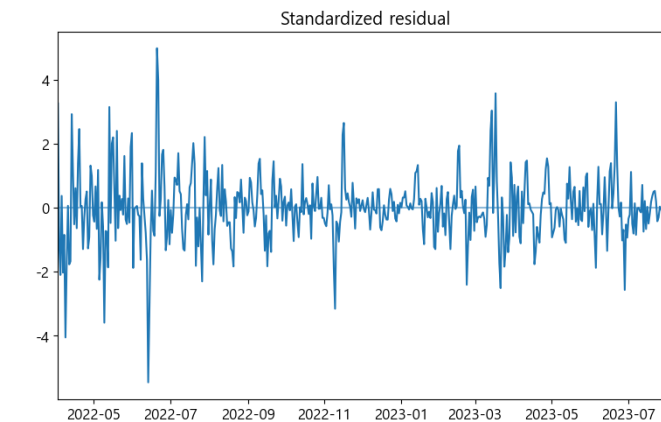
분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

```
model.plot_diagnostics(figsize=(20, 12))
plt.show()
```



## 학습 결과에 대한 예측값

```
fv = model.fittedvalues()
fv.head()
```

## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

```
2022-03-25 09:00:00    0.000000e+00
2022-03-26 09:00:00    5.582250e+07
2022-03-27 09:00:00    5.383184e+07
2022-03-28 09:00:00    5.498430e+07
2022-03-29 09:00:00    5.682831e+07
Freq: D, dtype: float64
```

## 향후 30일간의 시세 예측

```
fc = model.predict(n_periods=30)
fc.head()
```

```
2023-08-07 09:00:00    3.858298e+07
2023-08-08 09:00:00    3.865642e+07
2023-08-09 09:00:00    3.916127e+07
2023-08-10 09:00:00    3.904907e+07
2023-08-11 09:00:00    3.901622e+07
Freq: D, dtype: float64
```

## 시각화

```
plt.figure(figsize=(20,8))

# 원본 데이터
sb.lineplot(data=df, x=df.index, y='price', label='Original')
```



## 비트코인 가격 분석

## #01. 작업준비

패키지 참조

## #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

## #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

## #04. 데이터 검정

## #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

## #05. AutoARIMA 사용

```
# 원본에 대한 학습결과
sb.lineplot(x=fv.index, y=fv.values, label='FittedValues', linestyle='--')

# 예측 데이터
sb.lineplot(x=fc.index, y=fc.values, label='Predict', linestyle='--', color='red')

plt.xlabel('Month')
plt.ylabel('Price')
plt.legend()

# 그래프의 x축이 날짜로 구성되어 있을 경우 형식 지정
monthyearFmt = mdates.DateFormatter('%y.%m.%d')
plt.gca().xaxis.set_major_formatter(monthyearFmt)

plt.grid()
plt.show()
plt.close()
```

## 비트코인 가격 분석

### #01. 작업준비

패키지 참조

### #02 pyupbit 패키지 사용

조회 가능한 단위 목록

특정 단어를 포함하는 조회 가능 목록만 가져오기

현재 시세 가져오기

특정 기간에 대한 시세 데이터 가져오기

### #03. 데이터 전처리

시세가격에 대한 파생변수 추가

인덱스에 대한 날짜 형식 지정

### #04. 데이터 검정

### #04. ARIMA 분석

분석 모델 구축

분석 모델에 대한 예측치

향후 30일간의 데이터 예측

시각화

### #05. AutoARIMA 사용

