ARMA모형 모수 추정

다음과 같은 ARMA(p,q) 모형과 표본 시계열 자료가 있는 경우 최대 가능도 추정법으로 모수 ϕ_i, θ_i를 구할 수 있다.

$$Y_t + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} - \cdots + \theta_q \epsilon_{t-q}$$

MLE를 사용하기 위한 가능도 함수는 다음과 같다.

$$\begin{aligned} \mathcal{L}(\theta; \{y_j\}) &= \mathcal{L}(\phi_1, \cdots, \phi_p, \theta_1, \cdots, \theta_q; y_1, y_2, \cdots, y_N) \\ &= f_{Y_1,\cdots,Y_N}(y_1, y_2, \cdots, y_N; \phi_1, \cdots, \phi_p, \theta_1, \cdots, \theta_q) \end{aligned}$$

ARMA(p,q)의 가능도 함수

확률변수와 달리 확률과정 모형에서는 ϵ 값을 알아야 하기 때문에 다음처럼 반복적 방법을 사용한다.

(1)

우선 임의의 ϵ 값을 초기값으로 가정한다.

Y_t는 $Y_{t-1}, \cdots, Y_{t-p}, \epsilon_{t-1}, \cdots, \epsilon_{t-q}$ 에 의존하는 정규 분포 확률변수가 된다.

$$Y_t \sim -\phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \cdots - \phi_p Y_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

선형회귀방법으로 모수 $\phi_1, \cdots, \phi_p, \theta_1, \cdots, \theta_q$를 추정한다.

(2)

ϵ 값을 계산한다. 우선 다음과 같은 초기값 가정을 하자

$$\epsilon_p = \epsilon_{p-1} = \cdots = \epsilon_{p-q} = 0$$

그런 $\epsilon_{p+1}, \cdots, \epsilon_N$은 다음 공식에서 구할 수 있다.

$$\epsilon_{p+1} = Y_{p+1} + \phi_1 Y_p + \phi_2 Y_{p-1} + \cdots + \phi_p Y_1$$

$$\epsilon_{p+2} = Y_{p+2} + \phi_1 Y_{p+1} + \phi_2 Y_p + \cdots + \phi_p Y_2 - \theta_1 \epsilon_{p+1}$$

$$\vdots$$

이 ϵ 값을 이용하여 다시 (1)의 회귀분석을 실시하여 모수 $\phi_1, \cdots, \phi_p, \theta_1, \cdots, \theta_q$를 추정한다.

statsmodels 패키지를 사용한 ARMA 모수 추정

statsmodels 패키지는 ARMA 모수 추정을 위한 `ARMA` 라는 클래스와 이 클래스의 `fit` 메서드를 제공한다. `ARMA` 클래스의 인수로 시계열 표본과 차수를 넣어 인스턴스를 생성한 뒤 `fit` 메서드를 호출하면 추정 결과값을 가진 `ARMAResults` 클래스 자료를 반환한다. 다만 주의할 점은 **`ARMAResults` 에서는 AR계수 ϕ에 대해 `ArmaProcess` 클래스에서 정한 부호와 반대의 부호를 출력**한다.

ARMA 클래스와 몇가지 이론적 모형에서 시뮬레이션하여 나온 시계열 자료를 사용하여 모수를 측정하여 보자.

MA(1) 모수 추정의 예

$$Y_t = \epsilon_t + 0.9\epsilon_{t-1}$$

In [1]:

```
np.random.seed(0)
p = sm.tsa.ArmaProcess([1], [1, 0.9])
y = p.generate_sample(1000)
m = sm.tsa.ARMA(y, (0, 1))
r = m.fit()
print(r.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1000
Model:                     ARMA(0, 1)   Log Likelihood               -1405.255
Method:                       css-mle   S.D. of innovations              0.986
Date:                Mon, 01 Jul 2019   AIC                           2816.509
Time:                        20:29:27   BIC                           2831.233
Sample:                             0   HQIC                          2822.105

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0853      0.060     -1.429      0.153      -0.202       0.032
ma.L1.y        0.9174      0.013     70.513      0.000       0.892       0.943
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1           -1.0900           +0.0000j            1.0900            0.5000
-----------------------------------------------------------------------------
```

AR(1) 모수 추정의 예

$$Y_t = 0.9Y_{t-1} + e_t$$

In [2]:

```
np.random.seed(0)
p = sm.tsa.ArmaProcess([1, -0.9], [1])
y = p.generate_sample(1000)
m = sm.tsa.ARMA(y, (1, 0))
r = m.fit()
print(r.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1000
Model:                     ARMA(1, 0)   Log Likelihood               -1404.871
Method:                       css-mle   S.D. of innovations              0.985
Date:                Mon, 01 Jul 2019   AIC                           2815.742
Time:                        20:29:27   BIC                           2830.465
Sample:                             0   HQIC                          2821.337

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.4251      0.360     -1.181      0.238      -1.131       0.281
ar.L1.y        0.9144      0.013     72.007      0.000       0.890       0.939
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0936           +0.0000j            1.0936            0.0000
-----------------------------------------------------------------------------
```

AR(2) 모수 추정의 예

$$Y_t = 1.5Y_{t-1} - 0.75Y_{t-2} + \epsilon_t$$

In [3]:

```
np.random.seed(0)
p = sm.tsa.ArmaProcess([1, -1.5, 0.75], [1])
y = p.generate_sample(1000)
m = sm.tsa.ARMA(y, (2, 0))
r = m.fit()
print(r.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1000
Model:                     ARMA(2, 0)   Log Likelihood               -1406.159
Method:                       css-mle   S.D. of innovations              0.986
Date:                Mon, 01 Jul 2019   AIC                           2820.319
Time:                        20:29:27   BIC                           2839.950
Sample:                             0   HQIC                          2827.780
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1847      0.124     -1.484      0.138      -0.429       0.059
ar.L1.y        1.4866      0.021     69.795      0.000       1.445       1.528
ar.L2.y       -0.7371      0.021    -34.614      0.000      -0.779      -0.695
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0084           -0.5829j            1.1648           -0.0834
AR.2            1.0084           +0.5829j            1.1648            0.0834
-----------------------------------------------------------------------------
```

ARMA(1,1) 모수 추정의 예

$$Y_t = 0.6Y_{t-1} + \epsilon_t + 0.3\epsilon_{t-1}$$

In [4]:

```
np.random.seed(0)
p = sm.tsa.ArmaProcess([1, -0.6], [1, 0.3])
y = p.generate_sample(1000)
m = sm.tsa.ARMA(y, (1, 1))
r = m.fit()
print(r.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1000
Model:                     ARMA(1, 1)   Log Likelihood               -1405.147
Method:                       css-mle   S.D. of innovations              0.986
Date:                Mon, 01 Jul 2019   AIC                           2818.295
Time:                        20:29:28   BIC                           2837.926
Sample:                             0   HQIC                          2825.756

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1436      0.103     -1.400      0.162      -0.345       0.057
ar.L1.y        0.6175      0.034     18.294      0.000       0.551       0.684
ma.L1.y        0.2606      0.043      6.090      0.000       0.177       0.344
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.6195           +0.0000j            1.6195            0.0000
MA.1           -3.8370           +0.0000j            3.8370            0.5000
-----------------------------------------------------------------------------
```

실제 시계열 자료에 대한 ARMA 모형 계수 추정

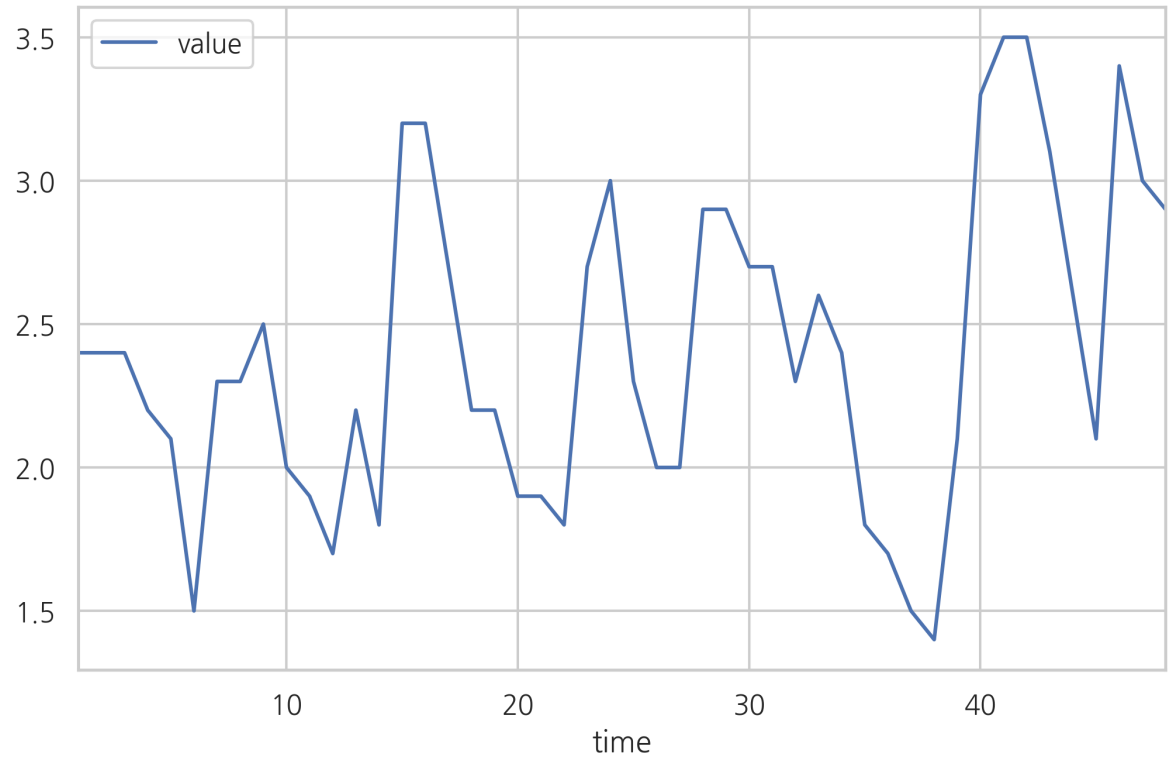
이번에는 실제 시계열 자료에 대해 ARMA 모형 계수를 추정해 보자.

황체형성 호르몬 모형 계수 추정

황체형성 호르몬의 모형 차수는 AR(1)로 추정했었다. 이 때 계수는 다음처럼 구한다.

In [5]:

```
data = sm.datasets.get_rdataset("lh")
df = data.data
df.plot(x="time", y="value")
plt.show()
```



이 시계열을 AR(1) 모형으로 보고 모수를 추정한 결과는 다음과 같다.

In [6]:

```
m = sm.tsa.ARMA(df.value, (1, 0))
r = m.fit()
print(r.summary())
```

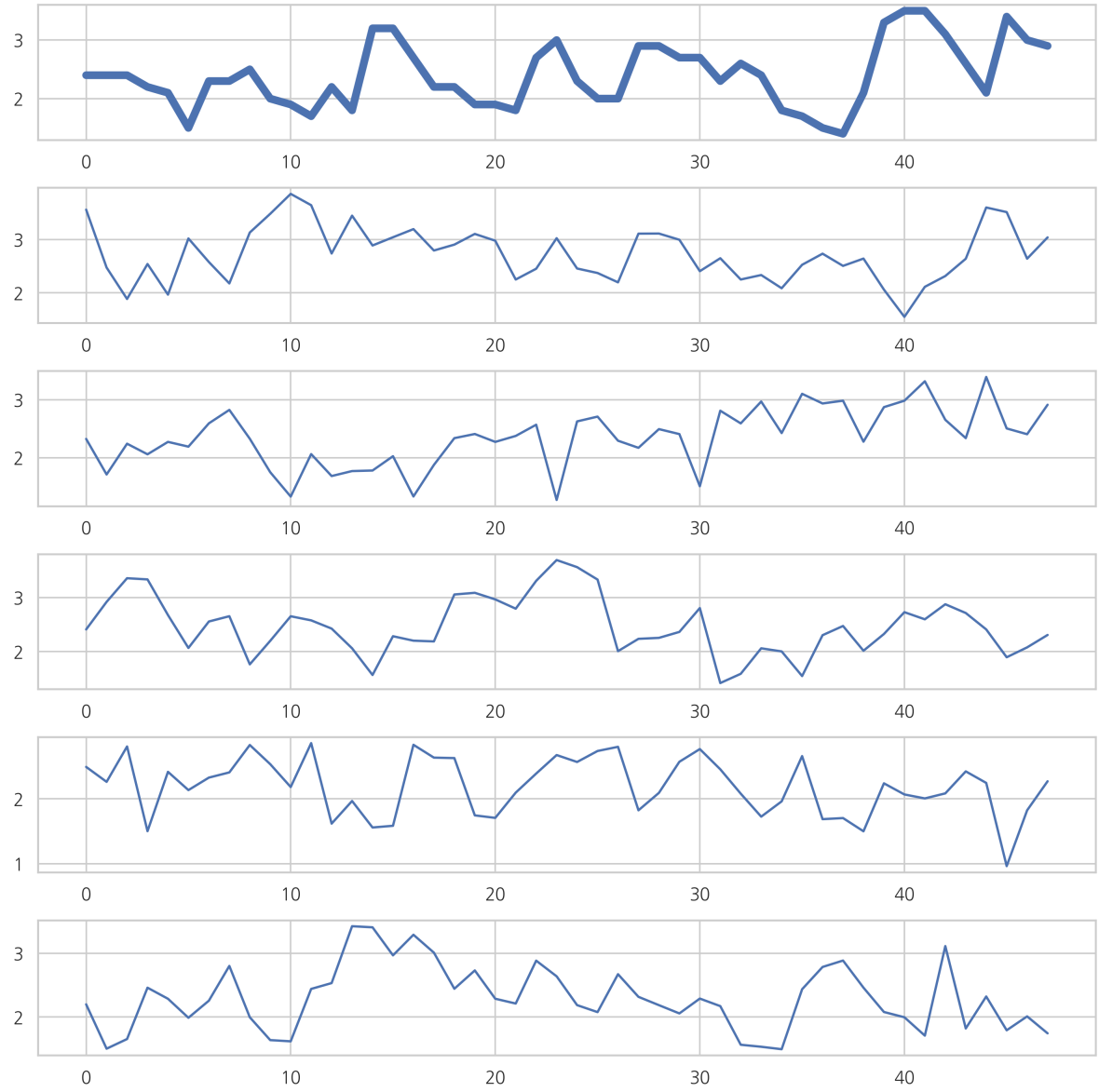
```
                              ARMA Model Results
==============================================================================
Dep. Variable:                  value   No. Observations:                   48
Model:                     ARMA(1, 0)   Log Likelihood                -29.379
Method:                       css-mle   S.D. of innovations              0.444
Date:                Mon, 01 Jul 2019   AIC                             64.758
Time:                        20:29:30   BIC                             70.372
Sample:                             0   HQIC                            66.880

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.4133      0.147     16.460      0.000       2.126       2.701
ar.L1.value    0.5739      0.116      4.939      0.000       0.346       0.802
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.7424           +0.0000j            1.7424            0.0000
-----------------------------------------------------------------------------
```

이 모수를 사용하여 다시 시뮬레이션한 결과는 다음과 같다. 굵은 실선이 원래 자료이다.

In [7]:

```
plt.figure(figsize=(10, 10))
np.random.seed(0)
p = sm.tsa.ArmaProcess(np.r_[1, -r.arparams], [1])
plt.subplot(6, 1, 1)
plt.plot(df.value, lw=5)
for i in range(5):
    plt.subplot(6, 1, i + 2)
    y = p.generate_sample(len(df), burnin=100) * np.sqrt(r.sigma2) + r.params[0]
    plt.plot(y)
plt.tight_layout()
plt.show()
```



MA(1) 모형으로 추정한 결과는 다음과 같다. AR(1) 모형보다 성능이 좋지 않음을 알 수 있다.

In [8]:

```
m = sm.tsa.ARMA(df.value, (0, 1))
r = m.fit()
print(r.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                  value   No. Observations:                   48
Model:                     ARMA(0, 1)   Log Likelihood                -31.052
Method:                       css-mle   S.D. of innovations              0.461
Date:                Mon, 01 Jul 2019   AIC                             68.104
Time:                        20:29:31   BIC                             73.717
Sample:                             0   HQIC                            70.225
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.4050      0.098     24.576      0.000       2.213       2.597
ma.L1.value    0.4810      0.094      5.093      0.000       0.296       0.666
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1           -2.0790           +0.0000j            2.0790            0.5000
-----------------------------------------------------------------------------
```

ARMA(1,1)모형으로 추정하면 가능도는 높지만 AIC, BIC가 나쁘다.

In [9]:

```
m = sm.tsa.ARMA(df.value, (1, 1))
r = m.fit()
print(r.summary())
```

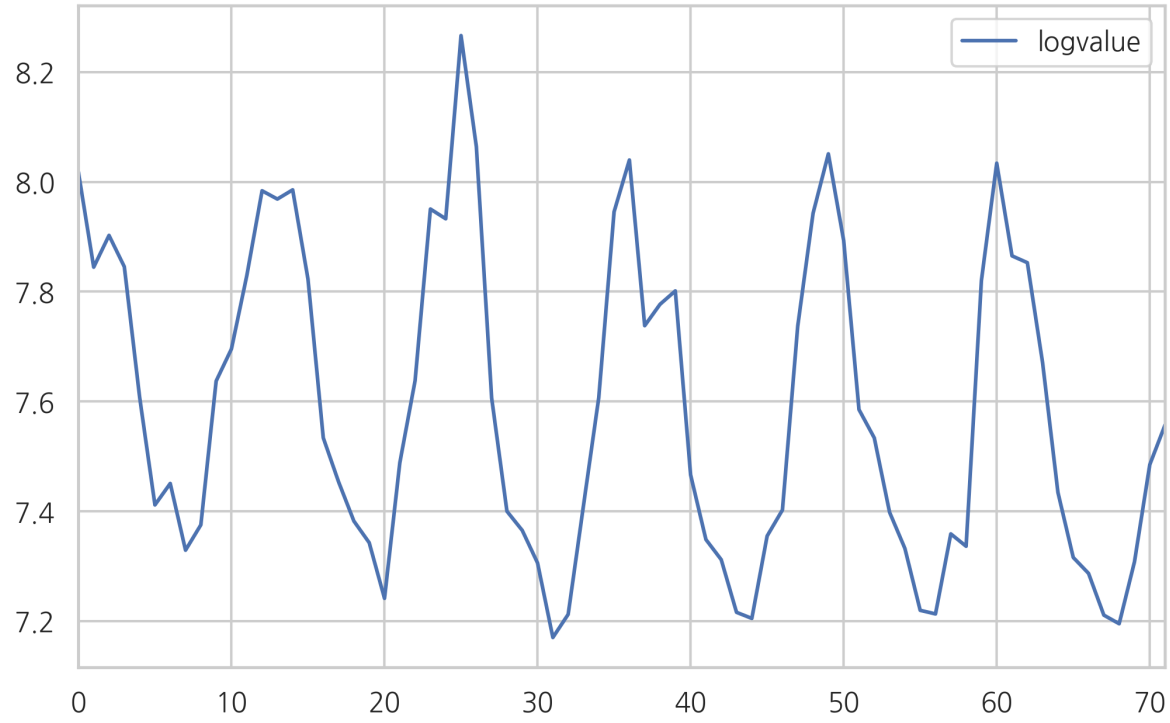
```
                              ARMA Model Results
==============================================================================
Dep. Variable:                  value   No. Observations:                   48
Model:                     ARMA(1, 1)   Log Likelihood                -28.762
Method:                       css-mle   S.D. of innovations              0.439
Date:                Mon, 01 Jul 2019   AIC                             65.524
Time:                        20:29:32   BIC                             73.009
Sample:                             0   HQIC                            68.353

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.4101      0.136     17.754      0.000       2.144       2.676
ar.L1.value    0.4522      0.177      2.556      0.014       0.105       0.799
ma.L1.value    0.1982      0.171      1.162      0.251      -0.136       0.532
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            2.2114           +0.0000j            2.2114            0.0000
MA.1           -5.0462           +0.0000j            5.0462            0.5000
-----------------------------------------------------------------------------
```

호흡기질환 사망자수

In [10]:

```
data = sm.datasets.get_rdataset("deaths", "MASS")
df = data.data
df["logvalue"] = np.log(df.value)
df.plot(y="logvalue")
plt.show()
```



이 시계열을 AR(2) 모형으로 보고 모수를 추정한 결과는 다음과 같다.

In [11]:

```
m = sm.tsa.ARMA(df.logvalue, (2, 0))
r = m.fit()
print(r.summary())
```

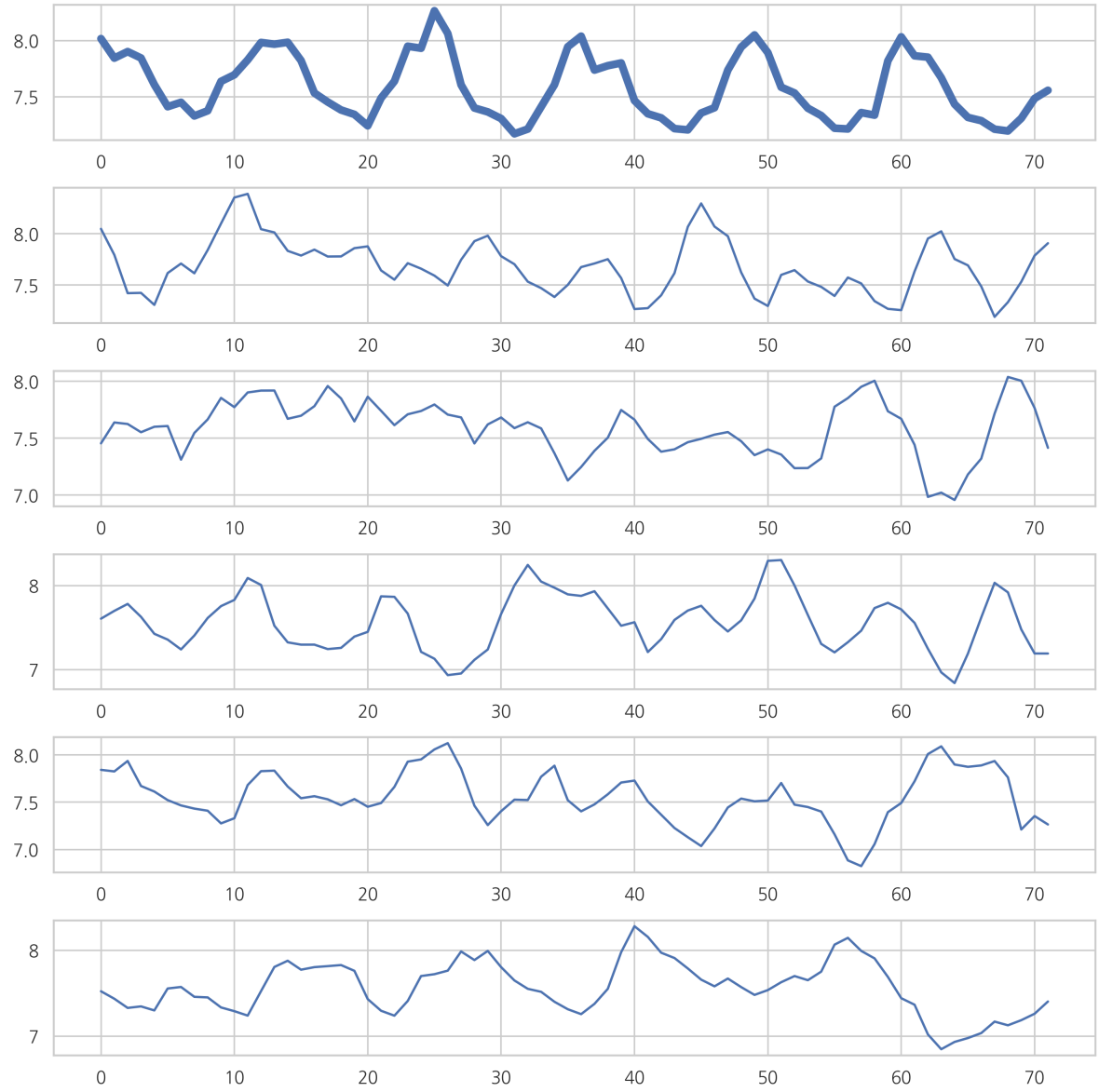
```
                              ARMA Model Results
==============================================================================
Dep. Variable:               logvalue   No. Observations:                   72
Model:                     ARMA(2, 0)   Log Likelihood                35.739
Method:                       css-mle   S.D. of innovations            0.146
Date:                Mon, 01 Jul 2019   AIC                          -63.478
Time:                        20:29:33   BIC                          -54.372
Sample:                             0   HQIC                         -59.853

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          7.5970      0.057    133.468      0.000       7.485       7.709
ar.L1.logvalue  1.2123      0.101     12.046      0.000       1.015       1.410
ar.L2.logvalue -0.5117      0.101     -5.090      0.000      -0.709      -0.315
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.1845           -0.7423j            1.3979           -0.0891
AR.2            1.1845           +0.7423j            1.3979            0.0891
-----------------------------------------------------------------------------
```

이 모수를 사용하여 다시 시뮬레이션한 결과는 다음과 같다. 굵은 실선이 원래 자료이다.

In [12]:

```
plt.figure(figsize=(10, 10))
np.random.seed(0)
p = sm.tsa.ArmaProcess(np.r_[1, -r.arparams], [1])
plt.subplot(6, 1, 1)
plt.plot(df.logvalue, lw=5)
for i in range(5):
    plt.subplot(6, 1, i + 2)
    y = p.generate_sample(len(df), burnin=100) * np.sqrt(r.sigma2) + r.params[0]
    plt.plot(y)
plt.tight_layout()
plt.show()
```



AR(1,0), AR(1,1), AR(2,0), AR(2,1), AR(1,2), AR(2,2) 모형을 사용한 결과는 다음과 같다.

In [13]:

```
from itertools import product

result = []
for p, q in product(range(3), range(3)):
    if (p == 0 & q == 0):
        continue
    m = sm.tsa.ARMA(df.logvalue, (p, q))
    try:
        r = m.fit()
        result.append({"p": p, "q": q, "LLF": r.llf, "AIC": r.aic, "BIC": r.bic})
    except:
        pass

pd.DataFrame(result)
```

Out[13]:

	AIC	BIC	LLF	p	q
0	-43.788594	-36.958595	24.894297	1	0
1	-55.965203	-46.858539	31.982602	1	1
2	-55.193694	-43.810363	32.596847	1	2
3	-63.478243	-54.371579	35.739122	2	0
4	-79.025760	-67.642430	44.512880	2	1

In [14]:

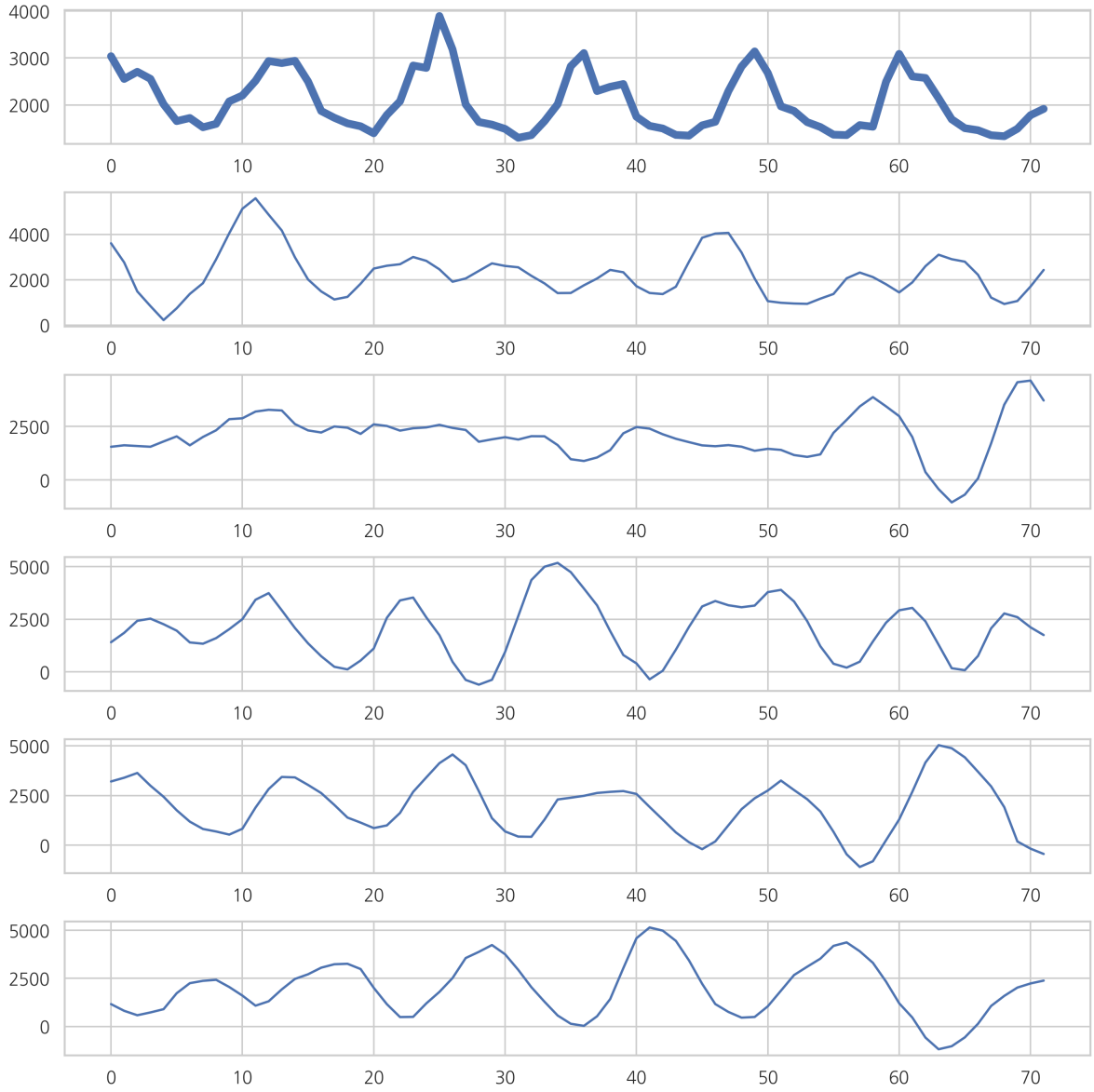
```
m = sm.tsa.ARMA(df.value.astype(float), (2, 1))
r = m.fit()
print(r.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                  value   No. Observations:                   72
Model:                     ARMA(2, 1)   Log Likelihood                -516.137
Method:                       css-mle   S.D. of innovations            309.351
Date:                Mon, 01 Jul 2019   AIC                           1042.275
Time:                        20:29:35   BIC                           1053.658
Sample:                             0   HQIC                          1046.806

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       2065.9330     42.605     48.490      0.000    1982.428    2149.438
ar.L1.value    1.6097      0.067     24.173      0.000       1.479       1.740
ar.L2.value   -0.8503      0.061    -13.842      0.000      -0.971      -0.730
ma.L1.value   -0.7315      0.076     -9.639      0.000      -0.880      -0.583
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.9466           -0.5293j            1.0845           -0.0811
AR.2            0.9466           +0.5293j            1.0845            0.0811
MA.1            1.3670           +0.0000j            1.3670            0.0000
-----------------------------------------------------------------------------
```

In [15]:

```
plt.figure(figsize=(10, 10))
np.random.seed(0)
p = sm.tsa.ArmaProcess(np.r_[1, -r.arparams], [1])
plt.subplot(6, 1, 1)
plt.plot(df.value, lw=5)
for i in range(5):
    plt.subplot(6, 1, i + 2)
    y = p.generate_sample(len(df), burnin=100) * np.sqrt(r.sigma2) + r.params[0]
    plt.plot(y)
plt.tight_layout()
plt.show()
```



결정론적 Seasonality 모형

In [16]:

```
def yearfraction2datetime(yearfraction, startyear=0):
    import datetime
    import dateutil
    year = int(yearfraction) + startyear
    month = int(round(12 * (yearfraction - year)))
    delta = dateutil.relativedelta.relativedelta(months=month)
    date = datetime.datetime(year, 1, 1) + delta
    return date

df["datetime"] = df.time.map(yearfraction2datetime)
df["month"] = df.datetime.dt.month

result = sm.OLS.from_formula('logvalue ~ C(month) + time + 0', data=df).fit()
y_seasonal = result.fittedvalues
y_nonseasonal = df.logvalue - y_seasonal
```

In [17]:

```
from itertools import product

result = []
for p, q in product(range(3), range(3)):
    if (p == 0 & q == 0):
        continue
    m = sm.tsa.ARMA(y_nonseasonal, (p, q))
    try:
        r = m.fit()
        result.append({"p": p, "q": q, "LLF": r.llf, "AIC": r.aic, "BIC": r.bic})
    except:
        pass

pd.DataFrame(result)
```

Out[17]:

	AIC	BIC	LLF	p	q
0	-151.064548	-144.234550	78.532274	1	0
1	-153.188303	-144.081638	80.594151	1	1
2	-155.929051	-144.545721	82.964526	1	2
3	-155.380672	-146.274008	81.690336	2	0
4	-153.684707	-142.301376	81.842353	2	1
5	-154.211493	-140.551497	83.105747	2	2

In [18]:

```
m = sm.tsa.ARMA(y_nonseasonal, (2, 2))
r = m.fit()

plt.figure(figsize=(10, 10))
np.random.seed(0)
p = sm.tsa.ArmaProcess(np.r_[1, -r.arparams], [1])
plt.subplot(6, 1, 1)
plt.plot(df.logvalue, lw=5)
for i in range(5):
    plt.subplot(6, 1, i + 2)
    y2 = p.generate_sample(len(df), burnin=100) * np.sqrt(r.sigma2) + r.params[0]
    y = y_seasonal + y2
    plt.plot(y)
plt.tight_layout()
plt.show()
```