ARIMA 모형과 단위근 검정

시계열 Y_t을 차분(difference)한 결과로 만들어진 시계열 $\nabla Y_t = Y_t - Y_{t-1}$이 ARMA 모형을 따르면 원래의 시계열 Y_t를 **ARIMA(Autoregressive Integrated Moving Average) 모형**이라고 한다.

만약 d번 차분한 후에야 시계열 $\nabla^d Y_t$가 ARMA(p,q) 모형을 따른다면 적분 차수(order of integration)가 d인 ARIMA 모형으로 **ARIMA(p, d, q)**로 표기한다. $q = 0$인 경우에는 ARI(p,d), $p = 0$인 경우에는 IMA(d,q)로 표기한다.

단위근 특성

ARIMA(p, 1, q) 모형은 특성 방정식(characteristic equation)이 $x = 1$이라는 단위근(unit root)를 가진다. 이 특성으로 인해 확률 과정이 ARIMA(p, 1, q)에 해당하는지 확인하는 검정 방법을 통틀어 **단위근 검정(unit root test)**이라고도 한다.

단위근 특성은 다음과 같이 확인한다. ARIMA(p,1,q) 모형을 차분한 값을 $W_t = Y_t - Y_{t-1}$라고 하자. W_t는 ARMA(p,q) 모형이므로 다음과 같이 표현할 수 있다.

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

이를 다시 원 시계열 Y_t로 표현하면,

$$Y_t - Y_{t-1} + \phi_1 (Y_{t-1} - Y_{t-2}) + \phi_2 (Y_{t-2} - Y_{t-3}) + \cdots + \phi_p (Y_{t-p} - Y_{t-p-1}) = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta$$

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

위 모형에 대한 특성 방정식은

$$1 + (-1 + \phi_1) x + (\phi_2 - \phi_1) x^2 + \cdots + (\phi_p - \phi_{p-1}) x^p - \phi_p x^{p+1} = 0$$

이를 인수분해하면

$$(1 - x)(1 + \phi_1 x + \phi_2 x^2 + \cdots + \phi_p x^p) = 0$$

이다. 즉 특성 방정식이 단위근($x = 1$)을 가진다.

이번에는 차분한 ARMA(p,q)모형이 아니라 차분하지 않은 형태의 ARMA(p,q)모형

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

을 살펴보자.

이 모형의 특성방정식은

$$\phi_p x^p + \cdots + \phi_1 x - 1 = 0$$

이고 이 방정식의 해가 $x = 1$이므로 대입하면 다음과 같은 관계를 얻을 수 있다.

$$\sum_{i=1}^{p} \phi_i = 1$$

IMA(1,1) 모형

ARIMA 모형의 가장 단순한 형태인 IMA(1,1)은 다음과 같다.

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

이 식을 일반 선형 확률 과정으로 표현하면 다음과 같다.

$$Y_t = \epsilon_t + (1-\theta)\epsilon_{t-1} + (1-\theta)\epsilon_{t-2} + (1-\theta)\epsilon_{t-3} + \cdots$$

즉 과거의 백색 잡음을 누적(cumulation)한 것이라고 볼 수 있다.

IMA(1,1) 모형의 자기상관계수는 시차(lag)에 상관없이 거의 1이다. 즉, 정상과정처럼 시차가 증가해도 자기상관계수가 감소하지 않는다.

$$\rho_l = \text{corr}[Y_t, Y_{t-l}] \approx 1$$

이렇게 자기상관계수가 빠르게 감소하지 않는 것이 ARIMA와 같은 적분 과정(integrated process)의 특징이다.

IMA(2,1) 모형

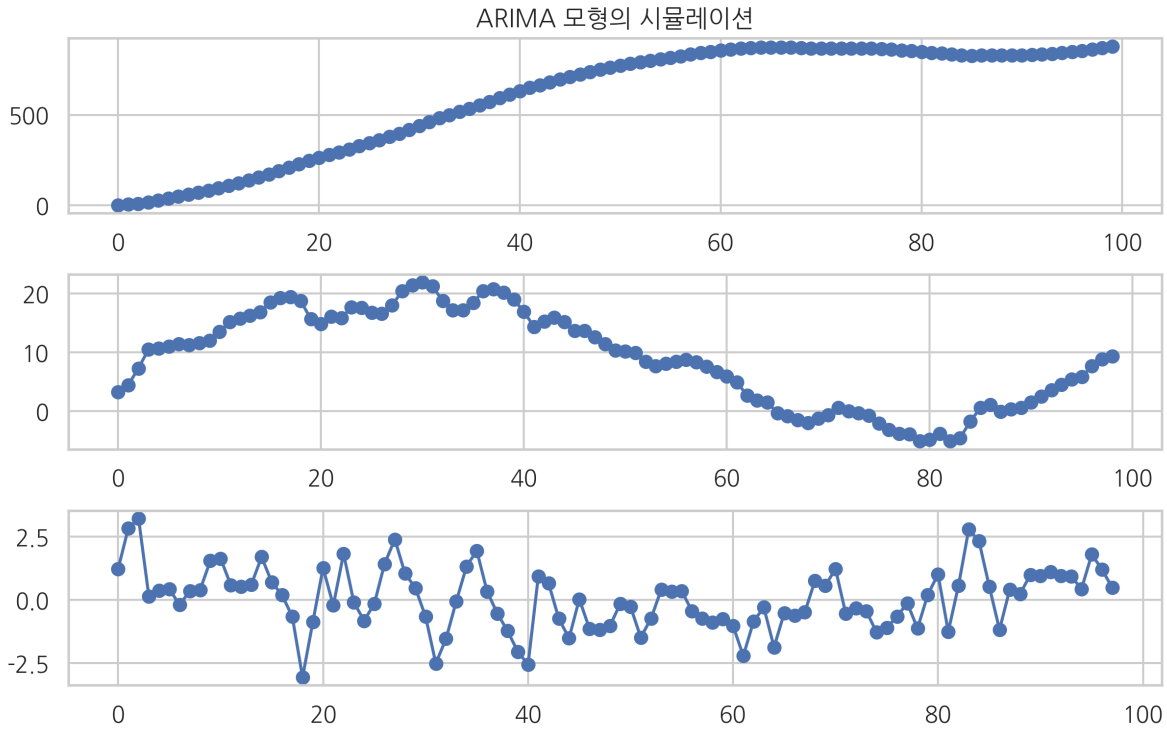
다음과 적분 차수가 2인 IMA(2,1)을 살펴보자

$$\nabla^2 Y_t = \epsilon_t - \theta\epsilon_{t-1}$$

이 모형을 시뮬레이션하여 ACF를 살펴보면 아래와 같다. statsmodels는 ARIMA 모형을 위한 별도의 클래스가 없기 때문에 `ArmaProcess` 를 사용한 후 누적합을 구하는 방식으로 시뮬레이션 한다.

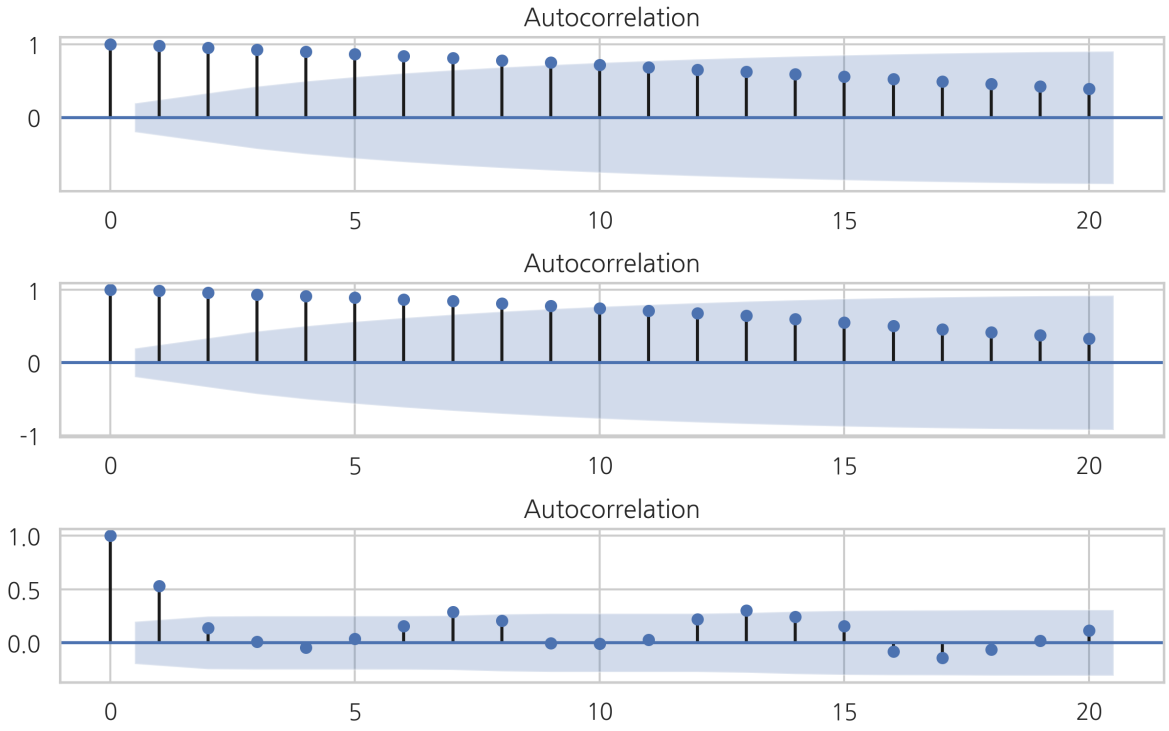
In [1]:

```
np.random.seed(0)
p = sm.tsa.ArmaProcess([1], [1, 0.6])
y2 = p.generate_sample(100).cumsum().cumsum()  # ARIMA d=2
y1 = np.diff(y2)
y0 = np.diff(y1)
plt.subplot(311)
plt.title("ARIMA 모형의 시뮬레이션")
plt.plot(y2, 'o-')
plt.subplot(312)
plt.plot(y1, 'o-')
plt.subplot(313)
plt.plot(y0, 'o-')
plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=1.0)
plt.show()
```



In [2]:

```
ax1 = plt.subplot(311)
sm.tsa.graphics.plot_acf(y2, ax=ax1)
ax2 = plt.subplot(312)
sm.tsa.graphics.plot_acf(y1, ax=ax2)
ax3 = plt.subplot(313)
sm.tsa.graphics.plot_acf(y0, ax=ax3)
plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=1.0)
plt.show()
```



과차분

시계열 자료를 원래의 적분 차수 이상으로 차분하는 것을 **과차분(over-differencing)**이라고 한다.

예를 들어 ARIMA(0,d,0) 모형의 시계열을 적분 차수에 맞게 차분하면 최종적으로 백색잡음이 된다. 만약 여기에서 추가적으로 차분을 하면 백색 잡음에 대한 MA(1) 모형이 되어 버린다. 한 번 더 차분한다면 이번에는 MA(2) 모형이 된다.

$$\nabla^d Y_t = \epsilon_t$$

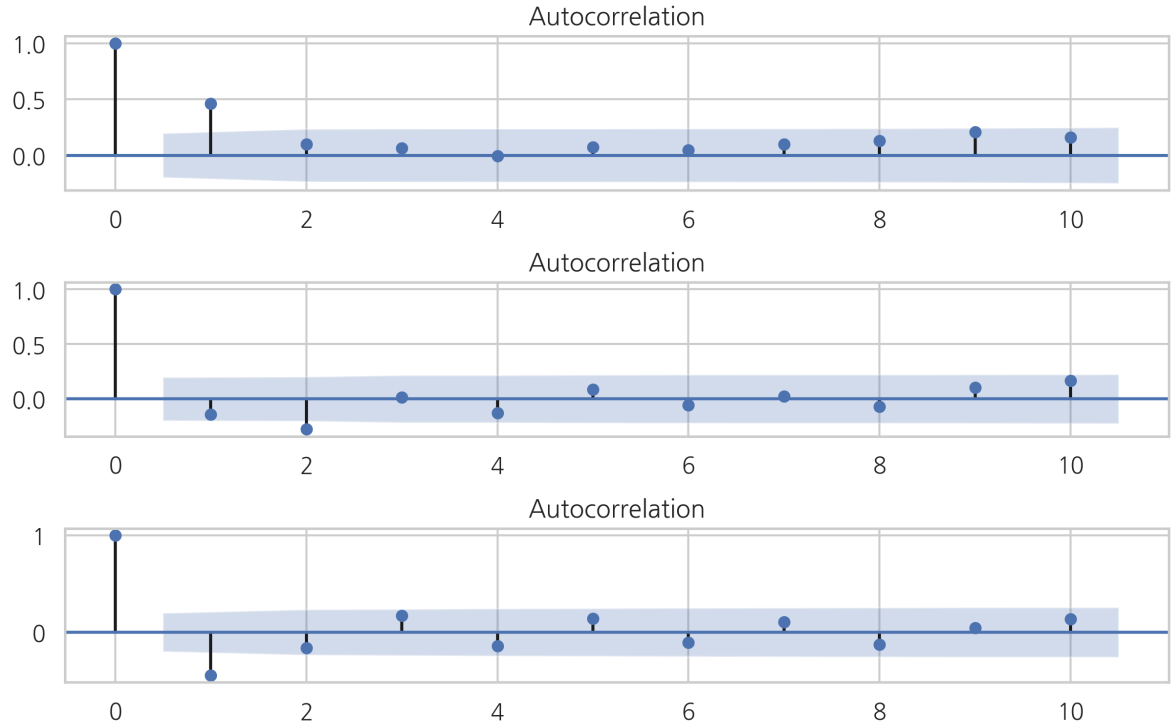
$$\nabla^{d+1} Y_t = \epsilon_t - \epsilon_{t-1}$$

$$\nabla^{d+2} Y_t = \epsilon_t - 2\epsilon_{t-1} + \epsilon_{t-2}$$

즉 모형 차수가 불필요하게 증가하게 되므로 ACF를 보면 큰 시차에 대한 자기상관계수값이 남아있는 것을 볼 수 있다.

In [3]:

```
np.random.seed(0)
p = sm.tsa.ArmaProcess([1], [1, 0.6])
y2 = p.generate_sample(100, burnin=100)
y1 = np.diff(y2)
y0 = np.diff(y1)
ax1 = plt.subplot(311)
sm.tsa.graphics.plot_acf(y2, lags=10,  ax=ax1)
ax2 = plt.subplot(312)
sm.tsa.graphics.plot_acf(y1, lags=10,  ax=ax2)
ax3 = plt.subplot(313)
sm.tsa.graphics.plot_acf(y0, lags=10,  ax=ax3)
plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=1.0)
plt.show()
```



Dickey-Fuller 단위근 검정

DF 검정(Dickey-Fuller 단위근 검정)의 회귀분석의 결과로 나타나는 가중치 계수를 검정통계량으로 사용한다.

시계열 Y_t이 AR(p)모형이면

$$Y_t + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} = \epsilon_t$$

이고 이 식을 정리하면

$$\begin{aligned} Y_t &= \sum_{i=1}^{p} \phi_i Y_{t-1} - \sum_{i=2}^{p} \phi_2 (Y_{t-1} - Y_{t-2}) - \cdots - \phi_p (Y_{t-p+1} - Y_{t-p}) + \epsilon_t \\ &= a_0 Y_{t-1} + a_1 \nabla Y_{t-1} + \cdots + a_p \nabla Y_{t-p} + \epsilon_t \end{aligned}$$

이다. 양변에서 Y_{t-1}을 빼면

$$\nabla Y_t = (a_0 - 1) Y_{t-1} + a_1 \nabla Y_{t-1} + a_2 \nabla Y_{t-2} + \cdots + a_p \nabla Y_{t-p} + \epsilon_t$$

을 얻을 수 있다.

따라서 ∇Y_t를 종속변수로, 원 시계열 Y_t와 차분값의 지연 값들 $\nabla Y_{t-1}, \nabla Y_{t-2}, \cdots, \nabla Y_{t-l}$을 독립변수로 회귀분석을 하면 $a_0, \ldots, a_p$를 구할 수 있다.

적분차수가 1이상이면

$$a_0 - 1 = \sum_{i=1}^{p} \phi_i - 1 = 0$$

이므로 이 값이 검정 통계량이 된다. 여기에서는 AR(p)모형을 사용하였지만 ARMA(p,q)모형에서도 백색 잡음을 변형하면 같은 결과를 얻을 수 있다.

DF 검정의 검정 통계량 a는 일반적인 검정과 달리 정규분포 등의 간단한 수식으로 나타나지 않고 미리 계산된 표를 사용하거나 근사식을 사용한다.

ADF 검정

Augmented Dickey-Fuller (ADF) 검정은 1차 누적에 의한 확률적 추세뿐 아니라 2차항으로 나타나는 결정론적 추세를 포함하는 시계열에 대해서 단위 근 검정을 할 수 있도록 DF 검정을 일반화 한 것이다.

ADF 검정도 DF 검정과 마찬가지로 회귀 분석 결과로 나온 계수를 검정 통계량으로 사용하며 회귀 방정식은 다음과 같다.

$$\nabla Y_t = \alpha + \beta t + (a_0 - 1) Y_{t-1} + a_1 \nabla Y_{t-1} + a_2 \nabla Y_{t-2} + \cdots + a_p \nabla Y_{t-p} + \epsilon_t$$

ADF 검정은 DF 검정을 포함하기 때문에 대부분의 통계 패키지는 ADF만을 지원한다.

statsmodels을 사용한 ADF 검정

statsmodels 패키지는 `statsmodels.tsa.adfuller` (http://statsmodels.sourceforge.net/stable/generated/statsmodels.tsa.stattools.adfuller.html) 라는 ADF 검정 함수를 제공한다.

입력 인수와 반환값은 다음과 같다.

- 입력 인수
  - x : 시계열 자료
  - maxlag : ADF 검정에 사용할 시차의 수. 디폴트 12*(nobs/100)^{1/4}

    - regression : 검정 사용할 모형.
        - 'nc': 결정론적 추세 없음. DF 검정
        - 'c': 상수항만 사용
        - 'ct': 상수항과 1차 추세 사용
        - 'ctt': 상수항과 1차 추세, 2차 추세 사용
    - autolag : 검정 AR 모형의 차수를 자동 결정하는 알고리즘 {'AIC', 'BIC', 't-stat', None}

- 반환값
    - adf : 검정 통계량
    - pvalue : MacKinnon(1994) 방식을 사용한 p-value 추정치
    - usedlag : 사용된 시차의 수
    - nobs : 분석에 자료의 수
    - critical values : 1%, 5%, 10% 수준에 해당되는 검정 통계량의 값

실제로 시뮬레이션으로 생성된 시계열 자로에 대해 ADF 검정을 실시해 본다.

In [4]:

```
p = sm.tsa.ArmaProcess([1], [-1, 0.6])
y2 = p.generate_sample(100, burnin=100).cumsum().cumsum()  # ARIMA d=2
y1 = np.diff(y2)
y0 = np.diff(y1)
```

위 코드에서 y2 는 적분차수가 2, y1 는 적분차수가 1, y0 는 적분차수가 0이므로 y2 , y1 는 $a_0 = 0$으로 귀무가설이 채택되어야 하고 y0 는 $a_0 \neq 0$으로 귀무가설이 기각되어야 한다. 다음 검정에서 이 결과를 확인할 수 있다.

In [5]:

```
sm.tsa.adfuller(y2)
```

Out[5]:

```
(2.9877347935825536,
 1.0,
 4,
 95,
 {'1%': -3.5011373281819504,
  '5%': -2.8924800524857854,
  '10%': -2.5832749307479226},
 230.25004856524797)
```

In [6]:

```
sm.tsa.adfuller(y1)
```

Out[6]:

```
(-0.8035452346311653,
 0.8180784629188602,
 7,
 91,
 {'1%': -3.50434289821397,
  '5%': -2.8938659630479413,
  '10%': -2.5840147047458037},
 234.16604597668746)
```

In [7]:

```
sm.tsa.adfuller(y0)
```

Out[7]:

```
(-5.839112781414058,
 3.8142611086678895e-07,
 6,
 91,
 {'1%': -3.50434289821397,
  '5%': -2.8938659630479413,
  '10%': -2.5840147047458037},
 230.8811374461522)
```