Solution for the worksheet
temporal time series analysis part II

Joaquin Rapela

April 27, 2025

1. Listing 12 shows the completed function *forecat* and Figure 1 shows the forecasting results for an AR(7) model.

```python
def forecast(x, acov, mu, m, max_h):
    Gamma_m = buildGamma(acov=acov, m=m)
    forecasts_means = np.empty(max_h, dtype=np.double)
    forecasts_vars = np.empty(max_h, dtype=np.double)
    xMinusMu_mR = (x-mu)[::-1][:m]
    for h in range(1, max_h+1):
        gamma_mh = acov[h:(h+m)]
        a_m = np.linalg.solve(Gamma_m, gamma_mh)
        forecasts_means[h-1] = mu + np.inner(a_m, xMinusMu_mR)
        forecasts_vars[h-1] = acov[0] - np.inner(a_m, gamma_mh)
    return forecasts_means, forecasts_vars
```

Listing 1: completed *forecast* function

2. Listing 8 shows the completed function *estimateCoefsAndNoisVarARpYW* and Figure 2 shows the estimated coefficients of an AR(3) model.

```python
def estimateCoefsAndNoiseVarARpYW(acov, p, N):
    Gammap = buildGamma(acov=acov, m=p)
    gammaph = acov[1:]
    phiHat = np.linalg.solve(Gammap, gammaph)
    sigma2Hat = acov[0] - np.inner(phiHat, gammaph)
    phiCovHat = sigma2Hat/N * np.linalg.inv(Gammap)
    return phiHat, phiCovHat, sigma2Hat
```

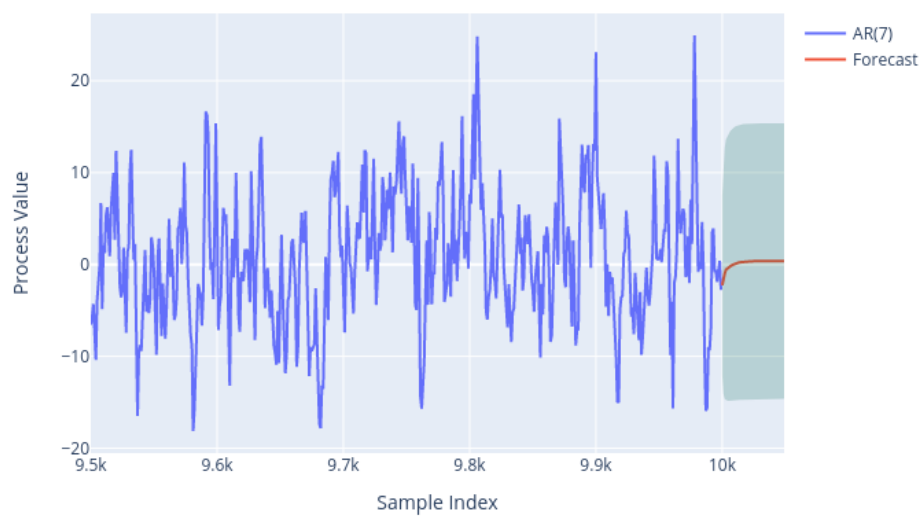Listing 2: completed *estimateCoefsAndNoisVarARpYW* function

Figure 1: Forecasting result for an AR(7) model
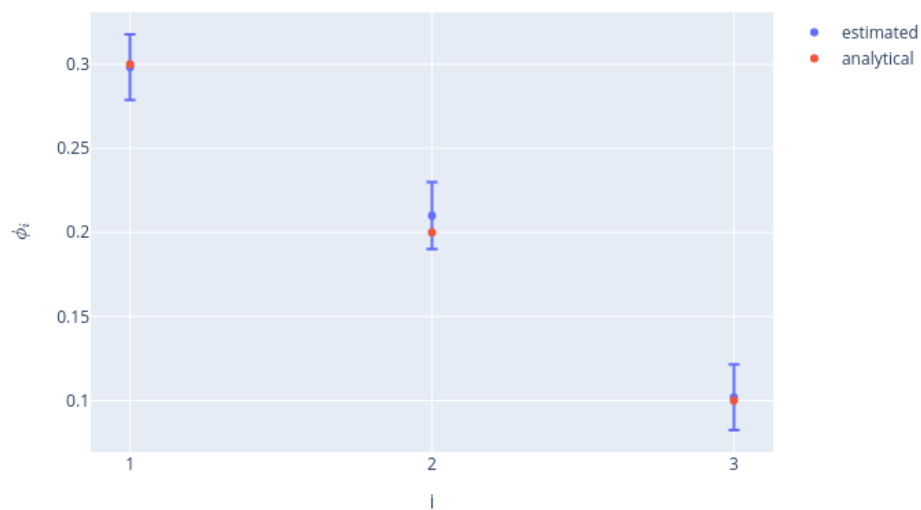
$\sigma^2 = 1.00, \hat{\sigma}^2 = 1.02$

Figure 2: True and estimated coefficients for an AR(3) model.