

Name David Plazas Escudero

Student code 201710005101

1. Let $\{2, 3, 1, -1, -4, -2, 0, 2, 1, -2\}$ be a sample of a given time series. Estimate $\hat{\rho}_1$, $\hat{\rho}_2$ and $\hat{\rho}_3$.

The autocorrelations are estimated computationally with the formula

$$\hat{\rho}_j = \frac{\hat{\gamma}_j}{\hat{\gamma}_0} \quad (1)$$

where

$$\hat{\gamma}_j = \frac{1}{n-1} \sum_{t=j+1}^n (x_t - \mu)(x_{t-j} - \mu).$$

The code used to estimate the first three autocorrelations of the given series is presented below:

```
import numpy as np

def autocov(x, j):
    s = 0
    mean = np.mean(x)
    n = np.size(x)
    for t in range(j, n):
        s += (x[t] - mean)*(x[t-j] - mean)
    return s/(n - 1)

def autocorr(x, j):
    return autocov(x, j)/autocov(x, 0)

x = [2, 3, 1, -1, -4, -2, 0, 2, 1, -2]
rs = []
for j in range(1, 4):
    rs.append(autocorr(x, j))
    print("rho_"+str(j), ":", autocorr(x, j))
```

The obtained autocorrelations are $\hat{\rho}_1 = 0.4545$, $\hat{\rho}_2 = -0.25$ and $\hat{\rho}_3 = 0. - 0.59$.

2. Using the sample from last exercise, calculate $\hat{\rho}_1^1$, $\hat{\rho}_2^2$ and $\hat{\rho}_3^3$. Hint: Use the Yule-Walker equations to estimate the AR models. The code to calculate the partial autocorrelations is presented below:

```
# returns vector of AR coefficients
def yule_walker(rs):
    j = rs.shape[0]
    A = np.ones((j, j))
    for k in range(j-1):
        A[(k+1):, k] = rs[:-(k+1), 0]
        A[k, (k+1):] = rs[:-(k+1), 0]
    return np.matmul(np.linalg.inv(A), rs)

# returns \hat{\rho}_j^j
def pacf(x, j, rs=None):
    if j == 1:
        return autocorr(x, j)
```

```

else:
    if rs is None:
        rs = []
        for i in range(1, j+1):
            rs.append(autocorr(x, i))
        rs = np.reshape(rs, (len(rs), 1))

    return yule_walker(rs)[-1, 0]

for j in range(1, 4):
    print('rho_' + str(j) + str(j) + ':', pacf(x, j))

```

and the estimated values are $\hat{\rho}_1^1 = 0.4545$, $\hat{\rho}_2^2 = -0.5755$ and $\hat{\rho}_3^3 = -0.2832$

3. Simulate the process $x_t = x_{t-1} + \epsilon_t - 0.6\epsilon_{t-1}$. Take the first difference and make the plot for the standard and partial autocorrelation functions of this first difference. What can be concluded from these plots? Note that

$$x_t = x_{t-1} + \epsilon_t - 0.6\epsilon_{t-1} \implies x_t - x_{t-1} = \epsilon_t - 0.6\epsilon_{t-1} \implies w_t = \epsilon_t - 0.6\epsilon_{t-1}.$$

Therefore, the first difference w_t is an MA(1) process with parameter $\theta_1 = 0.6$. Therefore, the autocorrelation plot must show a single spike (besides the 0-lag spike) exactly in the first lag, and the remaining autocorrelations should be close to 0. Furthermore, according to what was explained in class, the partial autocorrelation plot for an MA(q), particularly for $q = 1$, must decrease exponentially.

The code to simulate the process and obtain the autocorrelation and partial autocorrelation plots is presented below:

```

import scipy.stats as st
import matplotlib.pyplot as plt
import statsmodels.api as sm

plt.rc('text', usetex=True)
plt.rcParams.update({'font.size': 15})

def process(x0, T, theta, sigma2=1):
    e = st.norm.rvs(size=(T), loc=0, scale=sigma2**0.5)
    xs = np.zeros(T)
    xs[0] = x0
    for t in range(1, T):
        xs[t] = xs[t-1] + e[t] - theta*e[t-1]
    return xs

def diff(x):
    xs = np.zeros(x.size - 1)
    for t in range(xs.size):
        xs[t] = x[t+1] - x[t]
    return xs

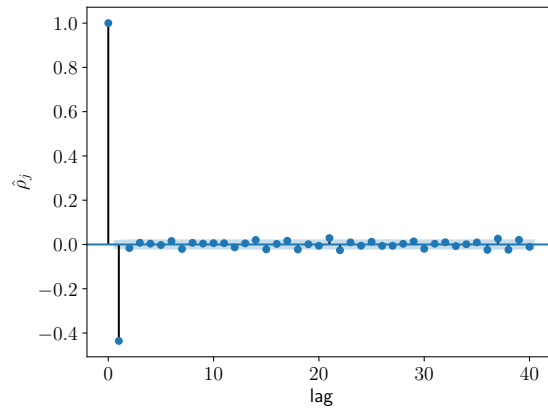
xs = process(0, 10000, 0.6)
ws = diff(xs)

lags = 40
fig1 = sm.graphics.tsa.plot_acf(ws, lags=lags)
plt.xlabel("lag")
plt.ylabel("$\hat{\rho}_j$")
plt.title("")
plt.savefig("acf.pdf", bbox_inches='tight')

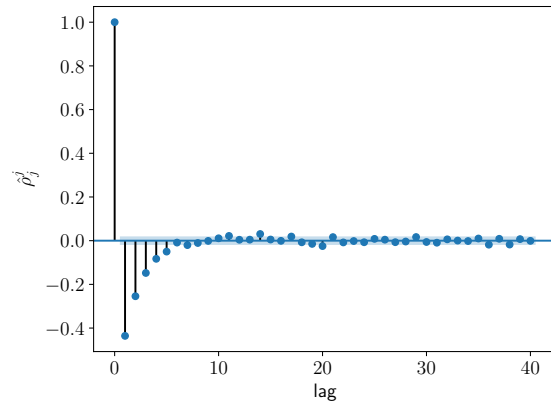
```

```
fig2 = sm.graphics.tsa.plot_pacf(ws, lags=lags)
plt.xlabel("lag")
plt.ylabel(" $\hat{\rho}_j$ ")
plt.title("")
plt.savefig("pacf.pdf", bbox_inches='tight')
```

The obtained plots for the autocorrelation and partial autocorrelation are presented in Figures 1a and 1b respectively. Note that these plots confirm the situation above described for an MA(1) process.



(a) Autocorrelations.



(b) Partial Autocorrelations.