MATH 545 - ASSIGNMENT 1

ASSIGNMENT

In the following questions, $\{Z_t\} \sim \text{WN}(0, \sigma^2)$, with $t \in \mathbb{Z}$, that is, we consider the process defined for all integers

1 Using the techniques established for the autoregressive model of order 1 (AR(1)), compute the autocorrelation function, $\rho_X(h), h = 0, \pm 1, \pm 2, \ldots$ for the following stationary processes:

(a) $\{X_t\}$ defined implicitly for all t by the equation

$$X_t + 0.1X_{t-1} = Z_t$$

2 MARKS

(b) $\{X_t\}$ defined implicitly for all t by the equation

$$X_t = 1.1X_{t-1} - 0.3X_{t-2} + Z_t$$

Hint: first find the representation

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j} \qquad t \in \mathbb{Z}$$

for constants ψ_j, $j = 0, 1, 2, \ldots$ using the recursive definition. 4 MARKS

(c) $\{X_t\}$ defined explicitly for all t by the equation

$$X_t = Z_t + Z_{t-1} - 0.5Z_{t-2}$$

3 MARKS

In each case, show the details of the derivation.

2 The *backshift* operator, B, is an operator that acts on time series processes $\{X_t\}$ such that for every t, $\text{B}X_t = X_{t-1}$. The operator has standard algebraic and arithmetic properties such that, for example, we can form polynomials in B: if $\Phi(\text{B})$ is such a polynomial of order $p \geq 1$, then

$$\Phi(\text{B})X_t \equiv \left(\sum_{j=0}^{p} \phi_j \text{B}^j \right) X_t = (\phi_0 + \phi_1 \text{B} + \phi_2 \text{B}^2 + \cdots + \phi_p \text{B}^p) X_t \equiv \sum_{j=0}^{p} \phi_j X_{t-j}.$$

for constants $\phi_0, \phi_1, \ldots, \phi_p$.

Show that the process $\{X_t\}$ from 1 (b) can be written in the form

$$\Phi(\text{B})X_t = Z_t$$

where $\Phi(z) = 1 + \phi_1 z + \phi_2 z^2$ is a polynomial of order two. Show that

$$\Phi(\text{B}) = (1 - \alpha_1 \text{B})(1 - \alpha_2 \text{B})$$

for two possibly complex values α_1, α_2. 5 MARKS

3 Using R, generate and plot realizations of length $n = 200$ from each of the processes in question 1, assuming that $Z_t \sim Normal(0, 1)$. Show the code you use, and provide the resulting plots.

6 MARKS

SOLUTIONS

1 (a) We may re-write

$$X_t = -0.1X_{t-1} + Z_t$$

and recognize this as an AR(1) process with parameter $\phi = -0.1$. Therefore using the recursion techniques from lectures (using the result that the process is stationary as given in the question) we have that

$$X_t = \sum_{j=0}^{\infty} \phi^j Z_{t-j}$$

and hence the process is zero mean, with

$$\gamma_X(h) = \mathbb{E}[X_t X_{t+h}] = \mathbb{E}\left[\left(\sum_{j=0}^{\infty} \phi^j Z_{t-j}\right)\left(\sum_{k=0}^{\infty} \phi^k Z_{t-k}\right)\right] = \mathbb{E}\left[\left(\sum_{j=0}^{\infty} \phi^j \phi^{j+h} Z_{t-j}^2\right)\right] = \sigma_Z^2 \phi^h \sum_{j=0}^{\infty} \phi^{2j}$$

as $\mathbb{E}[Z_{t-j} Z_{t+h-k}] = 0$ unless $k = j + h$. Thus, summing the geometric expression, we have

$$\gamma_X(h) = \frac{\sigma_Z^2 \phi^h}{1 - \phi^2}$$

and hence

$$\rho_X(h) = \frac{\gamma_X(h)}{\gamma_X(0)} = \phi^h = (-0.1)^h.$$

This is easily computed in R: the acf for $h = 0, 1, \ldots, 10$ are

```
rho<--0.1
rho^c(0:10)

+  [1]  1e+00 -1e-01  1e-02 -1e-03  1e-04 -1e-05  1e-06 -1e-07  1e-08 -1e-09
+ [11]  1e-10
```

2 MARKS

(b) Several different approaches are possible:

(i) **Using the infinite sum representation directly:** with the posited form of the solution

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}$$

for some constants $\psi_j, j \geq 0$, such that the summation is convergent in mean square (we know that the infinite sum converges in mean square as the process is given as stationary), we can substitute in to the AR formula to obtain

$$\sum_{j=0}^{\infty} \psi_j Z_{t-j} - \phi_1 \sum_{j=0}^{\infty} \psi_j Z_{t-j-1} - \phi_2 \sum_{j=0}^{\infty} \psi_j Z_{t-j-2} = Z_t$$

from which we can identify the coefficients, because the result must hold for all t. We have

Term	LHS	RHS
Z_t	ψ_0	1
Z_{t-1}	$\psi_1 - \phi_1\psi_0$	0
Z_{t-2}	$\psi_2 - \phi_1\psi_1 - \phi_2\psi_0$	0
Z_{t-3}	$\psi_3 - \phi_1\psi_2 - \phi_2\psi_1$	0
$\vdots$	$\vdots$	
Z_{t-j}	$\psi_j - \phi_1\psi_{j-1} - \phi_2\psi_{j-2}$	0
$\vdots$	$\vdots$	

Thus we have the recursion for $j \geq 2$

$$\psi_j = \phi_1\psi_{j-1} + \phi_2\psi_{j-2}$$

with $\psi_0 = 1$ and $\psi_1 = \phi_1$.

```
nlags<-1000; psi.vec<-numeric(nlags)
phi1<-1.1
phi2<--0.3
psi0<-1                                          #psi_0 = 1
psi.vec[1]<-1.1                                  #psi_1 = 1.1
psi.vec[2]<-phi1*psi.vec[1]+phi2*psi0
for(j in 3:nlags){
    psi.vec[j]<-phi1*psi.vec[j-1]+phi2*psi.vec[j-2]          #psi_r, r > 1
}
psi.vec[1:12]                                    #The first 12 psi values starting at psi_1
+  [1] 1.10000000 0.91000000 0.67100000 0.46510000 0.31031000 0.20181100
+  [7] 0.12889910 0.08124571 0.05070055 0.03139689 0.01932642 0.01183999
ARMAtoMA(ar = c(1.1,-0.3),lag.max=12)            #Check using the ARMAtoMA function
+  [1] 1.10000000 0.91000000 0.67100000 0.46510000 0.31031000 0.20181100
+  [7] 0.12889910 0.08124571 0.05070055 0.03139689 0.01932642 0.01183999
psi.vec[nlags]                                   #psi_{999} value
+ [1] 8.499662e-222
```

Then using the result (proved from first principles in lectures) that

$$\gamma_X(h) = \sigma_Z^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+h} \qquad \rho_X(h) = \frac{\sum_{j=0}^{\infty} \psi_j \psi_{j+h}}{\sum_{j=0}^{\infty} \psi_j^2}$$

we can compute the numerical acvs and acfs. Truncating the infinite sum by assuming that $\psi_j = 0$ for $j \geq 1000$ yields a computational formula for the covariances, we have:

```
acvs<-numeric(nlags)
psi0.vec<-c(psi0,psi.vec)
acvs[1]<-sum(psi0.vec^2)                          #h=0
for(i in 2:nlags){
    acvs[i]<-sum(psi0.vec[1:(nlags-i+1)]*psi0.vec[i:nlags])  #h=i
}
acfs<-acvs/acvs[1]
acvs[1:10]                                       #The first 10 acvs values
+  [1] 3.8690476 3.2738095 2.4404762 1.7023810 1.1404762 0.7438095 0.4760476
+  [8] 0.3005095 0.1877462 0.1163680
acfs[1:10]                                       #The first 10 acfs values
+  [1] 1.00000000 0.84615385 0.63076923 0.44000000 0.29476923 0.19224615
+  [7] 0.12304000 0.07767015 0.04852517 0.03007664
```

Note that the recursion above can be solved using linear recurrence relation methods: we have

$$\begin{bmatrix} \psi_j \\ \psi_{j-1} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \psi_{j-1} \\ \psi_{j-2} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix}^{j-1} \begin{bmatrix} \psi_1 \\ \psi_0 \end{bmatrix}$$

```
nlags<-1000; psi.vec<-numeric(nlags)
Phi<-matrix(c(phi1,1,phi2,0),2,2)
psi.mat<-matrix(0,ncol=nlags,nrow=2)
psi.mat[,1]<-c(1.1,1)                  #initialize the recursion
for(j in 2:nlags){
    psi.mat[,j]<-Phi %*% psi.mat[,j-1]       #apply the recursive formula
}
psi.mat[1,1:12]
+  [1] 1.10000000 0.91000000 0.67100000 0.46510000 0.31031000 0.20181100
+  [7] 0.12889910 0.08124571 0.05070055 0.03139689 0.01932642 0.01183999
```

(ii) **Using the recurrence relation:** we want to obtain X_t explicitly in terms of $\{Z_t\}$. We may use the recursion method to compute such a representation: to begin, we substitute in the equivalent expression for X_{t-1}

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t = \phi_1(\phi_1 X_{t-2} + \phi_2 X_{t-3} + Z_{t-1}) + \phi_2 X_{t-2} + Z_t$$

$$= \alpha_1^{(2)} X_{t-2} + \alpha_2^{(2)} X_{t-3} + \sum_{j=0}^{1} \psi_j Z_{t-j} \tag{1}$$

say, where $\psi_1 = 0, \psi_2 = \phi_1$ and

$$\alpha_1^{(2)} = \phi_1^2 + \phi_2 \qquad \alpha_2^{(2)} = \phi_1\phi_2$$

Let $\boldsymbol{\alpha}^{(2)} = (\alpha_1^{(2)}, \alpha_2^{(2)}, \ldots, \alpha_p^{(2)})^\top$ be the vector of coefficients. Then

$$\alpha_1^{(2)} = \phi_1\alpha_1^{(1)} + \alpha_2^{(1)} \qquad \alpha_2^{(2)} = \phi_2\alpha_1^{(1)}.$$

Therefore we may write $\boldsymbol{\alpha}^{(2)} = \mathbf{A}\boldsymbol{\alpha}^{(1)}$ for (2×2) matrix $\mathbf{A}$, where

$$\mathbf{A} = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix}$$

By repeatedly applying the recursion, we obtain

$$\boldsymbol{\alpha}^{(r)} = \mathbf{A}\boldsymbol{\alpha}^{(r-1)} = \mathbf{A}^2\boldsymbol{\alpha}^{(r-2)} = \cdots = \mathbf{A}^{r-1}\boldsymbol{\alpha}^{(1)}.$$

which defines the coefficients in the representation

$$X_t = \alpha_1^{(r)} X_{t-r} + \alpha_2^{(r)} X_{t-r-1} + \sum_{j=0}^{r-1} \psi_j Z_{t-j} \tag{2}$$

$$\boldsymbol{\alpha}^{(r)} = \mathbf{A}\boldsymbol{\alpha}^{(r-1)}$$

with $\psi_j = \alpha_1^{(j)}$. Diagonalizing $\mathbf{A}$, that is, writing

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \qquad \Longrightarrow \qquad \mathbf{A}^r = \mathbf{V}\mathbf{D}^r\mathbf{V}^{-1}$$

Now the eigenvalues can be found using the characteristic equation

$$\left| \begin{bmatrix} \phi_1 - \lambda & 1 \\ \phi_2 & -\lambda \end{bmatrix} \right| = \lambda(\lambda - \phi_1) - \phi_2 = 0$$

yielding the solutions $\left(\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}\right)/2$. Numerically, this yields the two eigenvalues $\lambda_1 = 0.6, \lambda_2 = 0.5$, which are both less than one in modulus, so we have that $\mathbf{A} \longrightarrow \mathbf{0}_{p,p}$ as required. Once we have computed

$$\boldsymbol{\alpha}^{(r)} = \mathbf{A}^{r-1}\boldsymbol{\alpha}^{(1)}$$

for each r, we can generate the ψ_j values by recalling that $\psi_j = \alpha_1^{(j)}$, and then using the formula above for the acvs and acfs. Using `R` for the numerical solution, with matrix $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} 1.1 & 1 \\ -0.3 & 0 \end{bmatrix}$$

we can check the eigen decomposition in `R`

```
A<-matrix(c(1.1,-0.3,1,0),2,2)
eigen(A)
+ eigen() decomposition
+ $values
+ [1] 0.6 0.5
+
+ $vectors
+            [,1]       [,2]
+ [1,]  0.8944272 -0.8574929
+ [2,] -0.4472136  0.5144958
```

indicating that the eigenvalues are 0.6, 0.5. Thus, $\mathbf{A}^r$ should converge to the zero matrix as $r \longrightarrow \infty$; we can check that also

```
A.r<-diag(c(1,1))                  #Identity matrix
for(r in 1:100){A.r<-A.r %*% A}    #A.r stores the A^r matrix
A.r
+               [,1]          [,2]
+ [1,]  3.919912e-22  6.533186e-22
+ [2,] -1.959956e-22 -3.266593e-22
```

This demonstrates that the process can be represented

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}$$

and that this process is well-defined (in mean-square terms). To compute the ψ values, we may use the recursive formula. We perform the calculation out to 1000 steps:

```
nlags<-1000; psi.vec<-numeric(nlags)
psi0<-1                                        #psi_0 = 1
psi.vec[1]<-1.1                                #psi_1 = 1.1
alpha.mat<-matrix(0,ncol=nlags,nrow=2)
alpha.mat[,1]<-c(1.1,-0.3)                  #initialize the recursion
for(r in 2:nlags){
    alpha.mat[,r]<-A %*% alpha.mat[,r-1]       #apply the recursive formula
    psi.vec[r]<-alpha.mat[1,r]                 #psi_r, r > 1
}
psi.vec[1:12]                                  #The first 12 psi values starting at psi_1
+  [1] 1.10000000 0.91000000 0.67100000 0.46510000 0.31031000 0.20181100
+  [7] 0.12889910 0.08124571 0.05070055 0.03139689 0.01932642 0.01183999
ARMAtoMA(ar = c(1.1,-0.3),lag.max=12)          #Check using the ARMAtoMA function
+  [1] 1.10000000 0.91000000 0.67100000 0.46510000 0.31031000 0.20181100
+  [7] 0.12889910 0.08124571 0.05070055 0.03139689 0.01932642 0.01183999
psi.vec[nlags]                                 #psi_{999} value
+ [1] 8.499662e-222
```

Once we have the representation in terms of ψ values, we may compute from first principles the autocovariances.

```
acvs<-numeric(nlags)
psi0.vec<-c(psi0,psi.vec)
acvs[1]<-sum(psi0.vec^2)                        #h=0
for(i in 2:nlags){
    acvs[i]<-sum(psi0.vec[1:(nlags-i+1)]*psi0.vec[i:nlags])  #h=i
}
acfs<-acvs/acvs[1]
acvs[1:10]                                   #The first 10 acvs values
+  [1] 3.8690476 3.2738095 2.4404762 1.7023810 1.1404762 0.7438095 0.4760476
+  [8] 0.3005095 0.1877462 0.1163680
acfs[1:10]                                   #The first 10 acfs values
```

```
+  [1] 1.00000000 0.84615385 0.63076923 0.44000000 0.29476923 0.19224615
+  [7] 0.12304000 0.07767015 0.04852517 0.03007664
```

We can check this result using the `ARMAacf` function in `R`

```
ARMAacf(ar=c(1.1,-0.3),lag.max=9)
+          0          1          2          3          4          5
+ 1.00000000 0.84615385 0.63076923 0.44000000 0.29476923 0.19224615
+          6          7          8          9
+ 0.12304000 0.07767015 0.04852517 0.03007664
```

Note that an analytic (closed-form) solution to the recurrence relation is possible using the eigenvector matrix $\mathbf{V}$ as we may write

$$\psi_j = \alpha_1^{(j)} = c_1\lambda_1^j + c_2\lambda_2^j \qquad j \geq 0$$

where c_1, c_2 are derived from the diagonalization result $\mathbf{A}^r = \mathbf{V}\mathbf{D}^r\mathbf{V}^{-1}$.

(iii) **Recursion on the acv:** using the technique from lectures (multiplying both sides by X_{t-h} and taking expectations), we have

$$\mathbb{E}[X_tX_{t-h}] = \phi_1\mathbb{E}[X_{t-1}X_{t-h}] + \phi_2\mathbb{E}[X_{t-2}X_{t-h}] + \mathbb{E}[Z_tX_{t-h}]$$

with $\phi_1 = 1.1$ and $\phi_2 = -0.3$. From this we conclude that

$$\gamma_X(h) = \phi_1\gamma_X(h-1) + \phi_2\gamma_X(h-2) \qquad h \geq 2.$$

as Z_t and X_{t-h} are uncorrelated by standard results. When $h = 0$, we have

$$\gamma_X(0) = \phi_1\gamma_X(-1) + \phi_2\gamma_X(-2) \equiv \phi_1\gamma_X(1) + \phi_2\gamma_X(2)$$

as we know $\gamma_X(-h) = \gamma_X(h)$, and when $h = 1$

$$\gamma_X(1) = \phi_1\gamma_X(0) + \phi_2\gamma_X(-1) \equiv \phi_1\gamma_X(0) + \phi_2\gamma_X(1).$$

From the latter equation we have

$$\gamma_X(1) = \frac{\phi_1}{1-\phi_2}\gamma_X(0).$$

Therefore, for each $h \geq 2$, for the acfs

$$\rho_X(h) = \phi_1\rho_X(h-1) + \phi_2\rho_X(h-2)$$

with $\rho_X(0) = 1, \rho_X(1) = \phi_1/(1-\phi_2)$. This recursion allows ready computation of the acfs.

```
acfs<-numeric(nlags)
acfs[1]<-1                                  #h=0
acfs[2]<-phi1/(1-phi2)                      #h=1
for(i in 3:nlags){
  acfs[i]<-  phi1*acfs[i-1] + phi2*acfs[i-2]   #h=i
}
acfs[1:10]                                  #The first 10 acfs values
+  [1] 1.00000000 0.84615385 0.63076923 0.44000000 0.29476923 0.19224615
+  [7] 0.12304000 0.07767015 0.04852517 0.03007664
```

Note that, once again, for an analytic solution, we have the linear recurrence relation

$$\begin{bmatrix} \rho_X(h) \\ \rho_X(h-1) \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \rho_X(h-1) \\ \rho_X(h-2) \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix}^{h-1} \begin{bmatrix} \rho_X(1) \\ \rho_X(0) \end{bmatrix}$$

and the same form of solution as above.

4 MARKS

(c) In this case we have the MA(2) process with coefficients $\theta_0 = \theta_1 = 1, \theta_2 = -0.5$. By direct calculation (or citing the result from lectures), we know that $\rho_X(h) = 0$ if $|h| > q = 2$. For the other three cases, we have

$$\gamma_X(h) = \begin{cases} \theta_0^2 + \theta_1^2 + \theta_2^2 & h = 0 \\ \theta_0\theta_1 + \theta_1\theta_2 & h = \pm 1 \\ \theta_0\theta_2 & h = \pm 2 \end{cases}$$

```
th0<-th1<-1
th2<--0.5
acvs<-numeric(4)
acvs[1]<-th0^2 + th1^2 + th2^2  #h=0
acvs[2]<-th0*th1 + th1*th2      #h=1
acvs[3]<-th0*th2                #h=2
acvs

+ [1]  2.25  0.50 -0.50  0.00

acfs<-acvs/acvs[1]
acfs

+ [1]  1.0000000  0.2222222 -0.2222222  0.0000000

ARMAacf(ma=c(1,-0.5),lag.max=3)

+          0          1          2          3
+  1.0000000  0.2222222 -0.2222222  0.0000000
```

Numerically, the three values are $\rho_X(0) = 1$, $\rho_X(1) = 0.2222222$ and $\rho_X(2) = -0.2222222$.

3 MARKS

In each case, show the details of the derivation.

2 By rearrangement of the implicit equation, we have that

$$X_t - 1.1X_{t-1} + 0.3X_{t-2} = Z_t$$

hence $\Phi(\mathrm{B}) = 1 - 1.1\mathrm{B} + 0.3\mathrm{B}^2$. Then we see that in fact

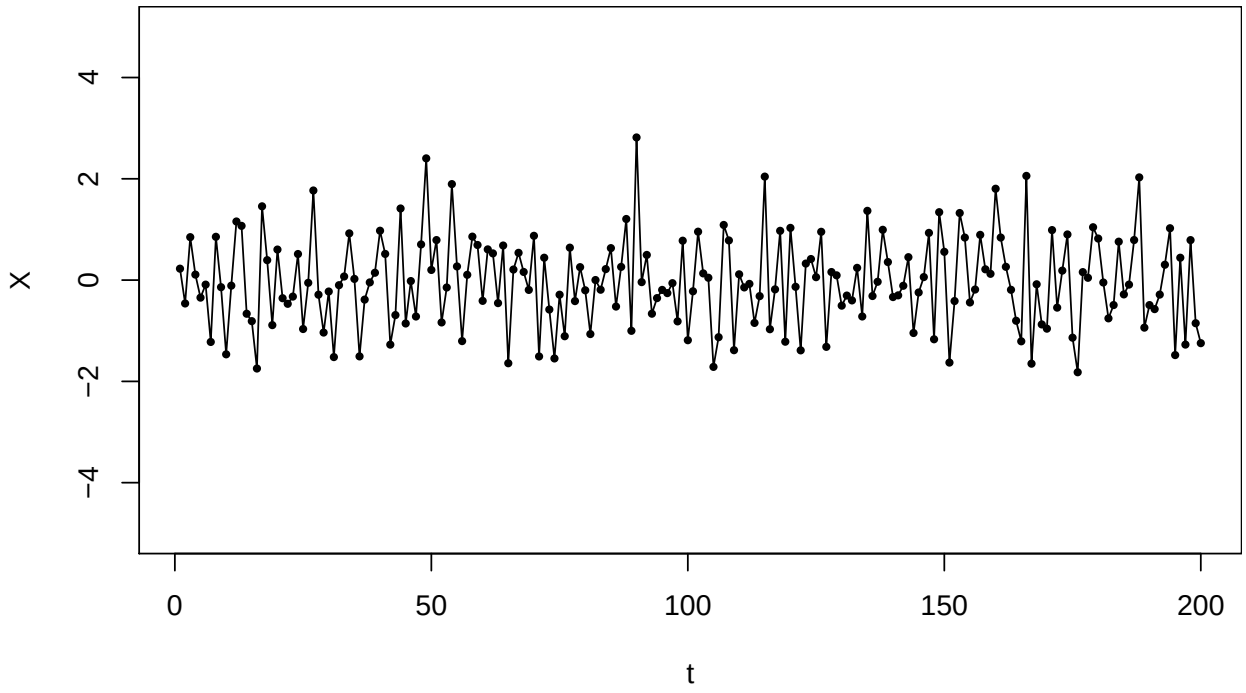
$$\Phi(\mathrm{B}) = (1 - 0.6\mathrm{B})(1 - 0.5\mathrm{B})$$

so $\alpha_1 = 0.6, \alpha_2 = 0.5$. Any quadratic polynomial can be factored into the product of two factors in this way, but here the α quantities are real-valued. Note that these two values match the eigenvalues from question 1b.

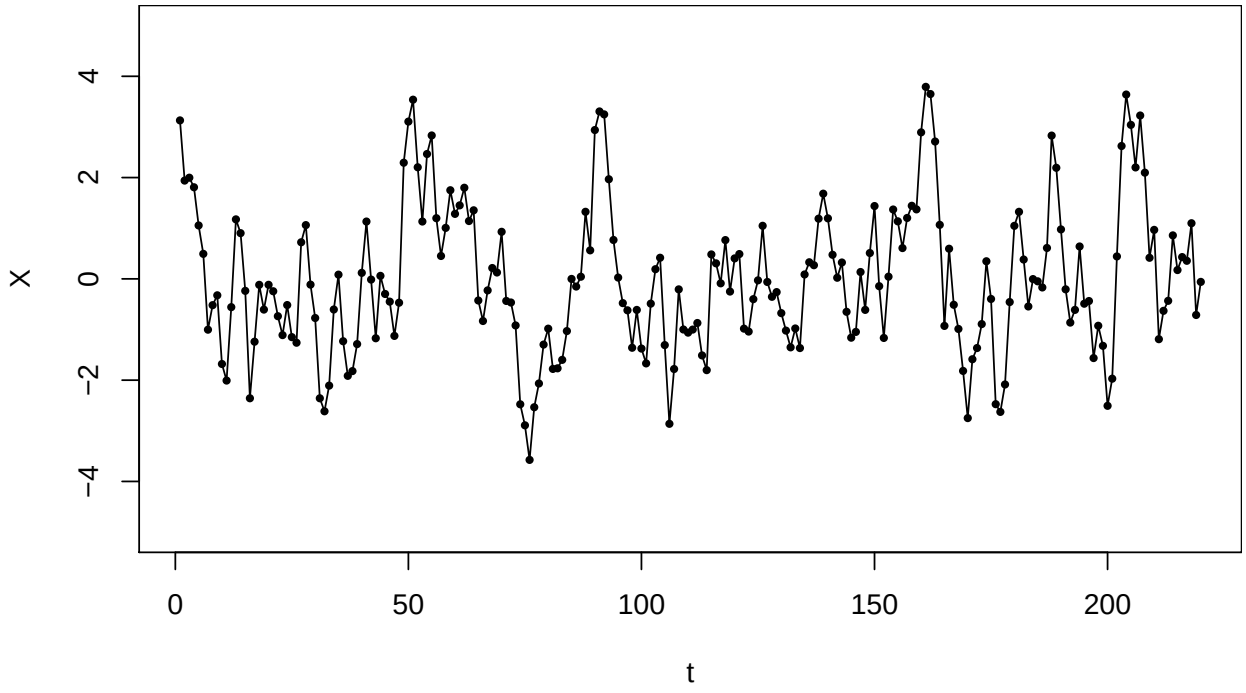
5 MARKS

3 For the three models, any sensible starting values can be chosen in the AR(p) case for 1 (a) and (b); for example, it is permissible here to set the initial values to zero (the expected value of X_t) or to randomly chosen $Z \sim Normal(0, 1)$ values. It is also possible (and in fact preferable) to generate X_t for $t \geq -20$ say, after initializing as mentioned. For (c), we can generate Z_{-2}, Z_{-1}, Z_0 and generate the X_t values accordingly.

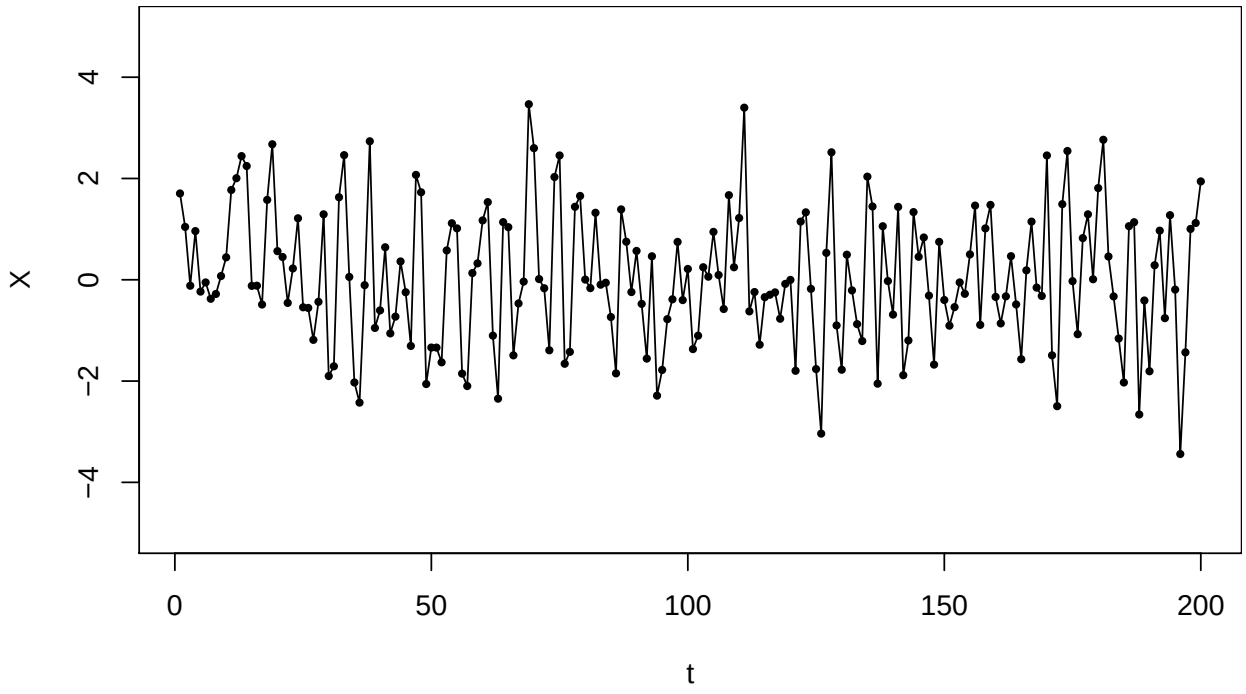
```
set.seed(2101)
n<-200;Z<-rnorm(n+20);X<-rep(0,n+20)
phi1<--0.1
X[1]<-0
for(i in 2:(n+20)){
    X[i]<-phi1*X[i-1]+Z[i]
}
X<-X[21:(n+20)]
par(mar=c(4,4,0,0))
plot(1:n,X,xlab='t',type='l',ylim=range(-5,5))
points(1:n,X,pch=19,cex=0.5)
```




```r
set.seed(2101)
n<-220;Z<-rnorm(n+20);X<-rep(0,n+20)
phi1<-1.1;phi2<--0.3
for(i in 3:(n+20)){
    X[i]<-phi1*X[i-1]+phi2*X[i-2]+Z[i]
}
X<-X[21:(n+20)]
par(mar=c(4,4,0,0))
plot(1:n,X,xlab='t',type='l',ylim=range(-5,5))
points(1:n,X,pch=19,cex=0.5)
```



```r
set.seed(2101)                    #Set the random number generator seed value
n<-200                            #Set the sample size
Z<-rnorm(n)                       #Generate Z[1],...Z[n]
Z.0<-rnorm(1)                     #Generate Z[0]
Z.1<-rnorm(1)                     #Generate Z[-1]
Z.extra<-c(Z.1,Z.0,Z)             #Concatenate
X<-Z.extra[3:(n+2)]+Z.extra[2:(n+1)]-0.5*Z.extra[1:n]   #Define the time series X
par(mar=c(4,4,0,0))
plot(1:n,X,xlab='t',type='l',ylim=range(-5,5))
points(1:n,X,pch=19,cex=0.5)
```



6 MARKS

Note: In this case, as the white noise process is Gaussian, we can in fact proceed using the fact that $\{X_t\}$ must also be Gaussian, with mean zero and variance covariance matrix given by the acvs. Therefore we can use the `mvrnorm` function in `R`: we also use the computational functions `ARMAtoMA` to obtain the linear process expansion and compute the variance of $\{X_t\}$, `ARMAacf` to compute the acf, and `toeplitz` to form the matrix.

```r
set.seed(2101)
library(MASS)
n<-200;
sigX1<-sum(ARMAtoMA(c(-0.1),lag.max=100)^2)
Sig1<-toeplitz(sigX1*ARMAacf(ar=c(-0.1), lag.max = 199))
X<-mvrnorm(n,mu=rep(0,n),Sigma=Sig1)
sigX2<-sum(ARMAtoMA(c(1.1,-0.3),lag.max=100)^2)
Sig2<-toeplitz(sigX2*ARMAacf(ar=c(1.1,-0.3), lag.max = 199))
X<-mvrnorm(n,mu=rep(0,n),Sigma=Sig2)
sigX3<-sum(c(1,1,-0.5)^2)
Sig3<-toeplitz(sigX3*ARMAacf(ma=c(1,-0.5), lag.max = 199))
X<-mvrnorm(n,mu=rep(0,n),Sigma=Sig3)
```