The University of British Columbia

Computer Science/Data Science 405/505 Modelling and Simulation

Assignment 4 Solutions

1. 100 insects are placed in a container holding a certain amount of insecticide. After 1 hour, 44 insects have died. Assuming that the insects survive independently of each other, use a binomial distribution model, together with maximum likelihood, to estimate the probability that more than 50 insects would survive in another experiment held under identical conditions.

   With p defined as the probability of an insect surviving, the likelihood function is

$$L(p) = p^x(1-p)^{n-x}$$

   where x is the number of insects that survive, in an experiment with n insects. Differentiating $log(L(p))$ with respect to p, gives

$$\frac{x}{p} - \frac{n-x}{1-p}$$

   Setting this to 0 and solving for p, we have $\widehat{p} = x/n$. In the experiment that was conducted, 56 insects survived, out of 100, so $\widehat{=}$.56. We can use this maximum likelihood estimate to estimate the probability that more than 50 insects survive in another experiment as follows:

```
1 - pbinom(50, 100, .56) # i.e. 1 - P(X <= 50)

## [1] 0.8659234
```

2. Data from the German stock exchange is in the DAX column of `EuStockMarkets`.

   (a) Store the successive differences of the log of the data in an object called `DAXlogreturn`.

```
DAX <- EuStockMarkets[, 1]
DAXlogreturn <- diff(log(DAX))
```

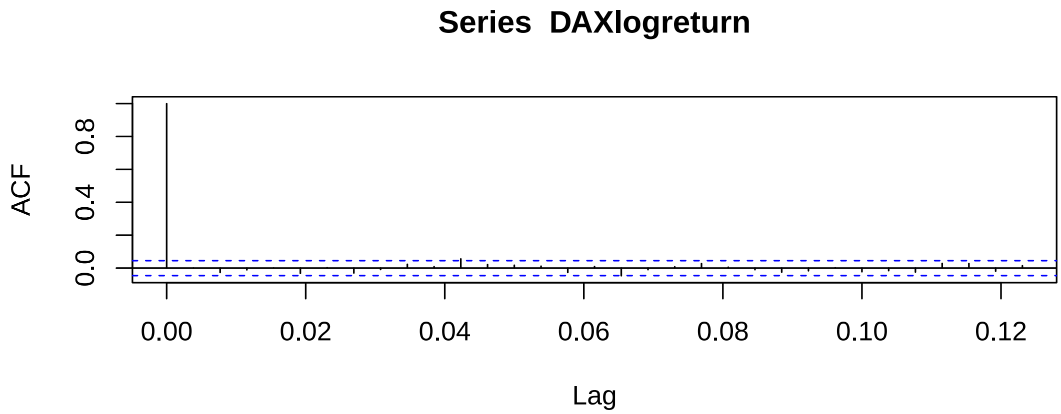
   (b) Calculate the mean of the log returns. This represents a deterministic drift in the series which translates into a deterministic trend - either upwards or downwards.

```
drift <- mean(DAXlogreturn)
drift

## [1] 0.0006520417
```

   (c) Apply the `acf()` function to these data. Is there evidence of linear dependence (i.e. autocorrelation)? If so, at which lags?

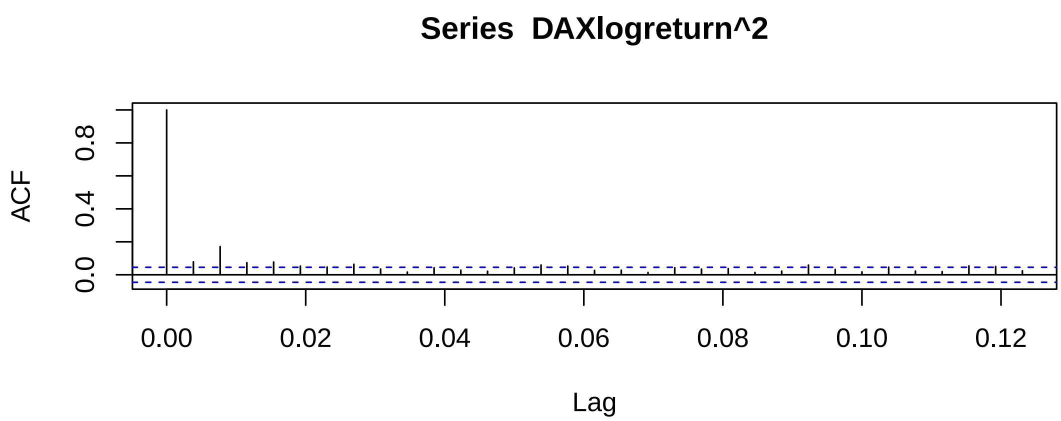
```
acf(DAXlogreturn)
```



There is no evidence of linear dependence in the data.

(d) Apply the `acf()` function to the squared data points. Is there evidence of autocorrelation now? If so, at which lags?

```
acf(DAXlogreturn^2)
```



There is evidence of dependence now, particularly at the 2nd lag.

(e) We can obtain approximate values to a_0, a_1 and a_2 for an ARCH(2) model by fitting an AR(2) model to the squared data points. Apply this technique to the `DAXlogreturn` data. Write out the fitted AR(2) model.

```
DAX.ar2 <- arima(DAXlogreturn^2, order = c(2, 0, 0))
DAX.ar2

##
## Call:
## arima(x = DAXlogreturn^2, order = c(2, 0, 0))
##
## Coefficients:
##          ar1     ar2  intercept
##       0.0658  0.1661      1e-04
## s.e.  0.0229  0.0229      0e+00
```

```
##
## sigma^2 estimated as 8.863e-08:  log likelihood = 12456.1,  aic = -24904.2
```

Let X_i be the ith squared log return. Then

$$(X_i - .0001) = .0658(X_{i-1} - .0001) + .1661(X_{i-2} - .0001) + \varepsilon$$

where ε has variance 8.86.

(f) Using the ϕ estimates from the fitted AR(2) model, write out an approximate ARCH model for the `DAXlogreturn` data.

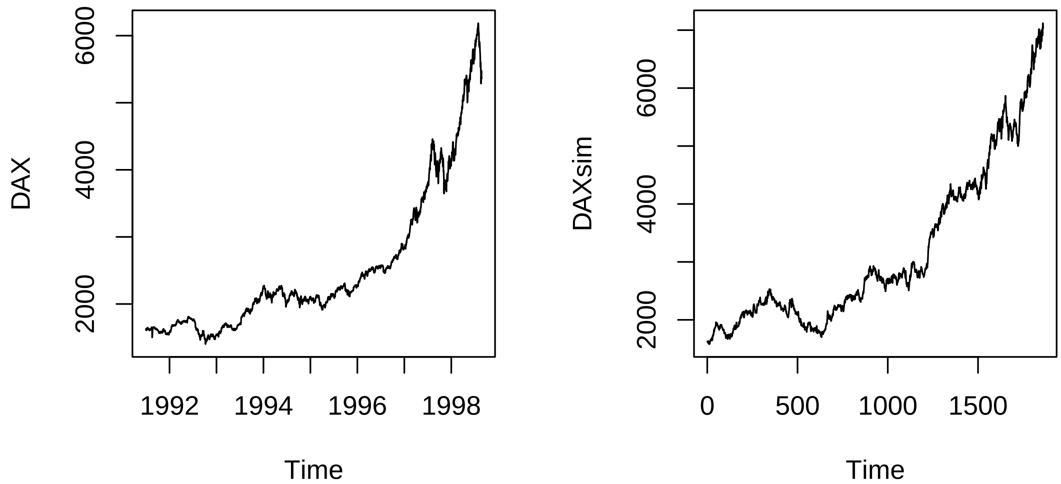
Let Y_i be the ith log return. Then

$$Y_i = \sqrt{.0001 + .0658Y_{i-1}^2 + .1661Y_{i-2}^2}Z_i$$

where Z_i is a standard normal random variable.

(g) Using the fitted model, simulate a time series of the same length as the DAX series which has essentially the same properties, and plot the result, together with the original data. Make sure to include the drift term in your model.

```
out <- arima(DAXlogreturn^2, order=c(2, 0, 0)) # (e)
phi <- out$coef[1:2];  xbar <- out$coef[3];  s2 <- out$sigma2
n <- length(DAX)
# Simulate from this model  # (f)
y <- numeric(n)
y[1:2] <- diff(log(DAX))[1:2] # starting values for process
Z <- rnorm(n) # standard normals used in ARCH
for (i in 3:n) {
    s <- sqrt(xbar + phi[1]*y[i-1]^2 + phi[2]*y[i-2]^2)*Z[i]
y[i] <- s
}
# y contains log returns, but an initial value is needed to
# re-accumulate the prices, and the drift term must be added in:
y <- c(log(DAX[1]), y + drift)
DAXsim <- exp(cumsum(y)) # simulated prices
par(mfrow=c(1, 2))  # (g) compare trace plots of real and simulated data.
ts.plot(DAX)
ts.plot(DAXsim)
```



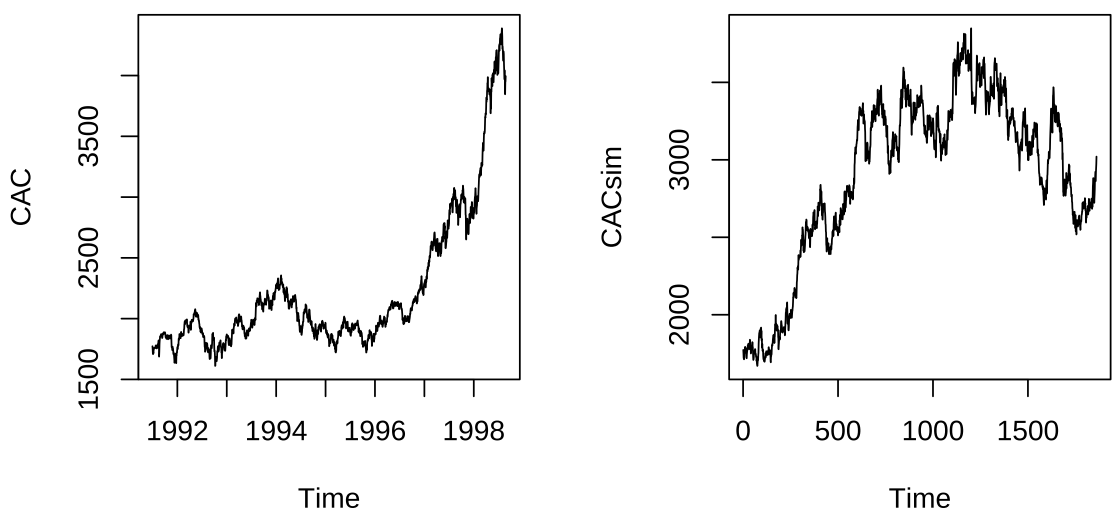
3. Fit an ARCH(2) model to the French CAC stock market data (the 3rd column of `EuStockMarkets`). Simulate a new series of the same length with the same properties as the CAC data, and plot the simulated data, as well as the original data.

```
set.seed(963621) # use this to reproduce the output below
CAC <- EuStockMarkets[, 3]
CAClogreturn <- diff(log(CAC))
out <- arima(CAClogreturn^2, order=c(2, 0, 0))
out # fitted ARCH(2) model

##
## Call:
## arima(x = CAClogreturn^2, order = c(2, 0, 0))
##
## Coefficients:
##          ar1     ar2  intercept
##        0.107  0.1107      1e-04
## s.e.   0.023  0.0230      0e+00
##
## sigma^2 estimated as 6.282e-08:  log likelihood = 12776.05,  aic = -25544.11

phi <- out$coef[1:2];  xbar <- out$coef[3];  s2 <- out$sigma2
n <- length(CAC)
# Simulate from this model
y <- numeric(n)
y[1:2] <- diff(log(CAC))[1:2] # starting values for process
Z <- rnorm(n) # standard normals used in ARCH
for (i in 3:n) {
    s <- sqrt(xbar + phi[1]*y[i-1]^2 + phi[2]*y[i-2]^2)*Z[i]
y[i] <- s
```

```
}
# y contains log returns, but an initial value is needed to
# re-accumulate the prices, and the drift term must be added in:
y <- c(log(CAC[1]), y + drift)
CACsim <- exp(cumsum(y)) # simulated prices
par(mfrow=c(1, 2))  # compare trace plots of real and simulated data.
ts.plot(CAC)
ts.plot(CACsim)
```



4. Containers are temporarily stored at a stockyard with capacity to store 3 containers. At the beginning of each day, precisely one container arrives at the stockyard, unless the stockyard is full; in that case, the container is taken elsewhere. Each container stays a certain amount of time before it is removed. The residency times of the containers are independent of each other. A container will be removed during a given day with probability $p = .8$ (independently of how many days the container has been stored).

   Let X_n denote the number of containers in the stockyard at the end of day n. $\{X_n\}$ is a Markov chain with state space $\{0, 1, 2, 3\}$.

   (a) Find the transition matrix.

   This is somewhat like the problem discussed in class, but now the length of time each container stays is a random quantity. Start by noting that if there are no containers at the end of a day, then the one container that arrives the next must leave with probability 0.8. When there is one container at the end of the day, and another arrives at the start of the next day, there must be a probability of $.8^2$ *that both containers are taken away, and a probability of* $.2^2$ *that both containers remain.*

$$P = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.64 & 0.32 & 0.04 & 0 \\ 0.512 & 0.384 & 0.096 & 0.008 \\ 0.512 & 0.384 & 0.096 & 0.008 \end{bmatrix}$$

When $X_n = 3$*, no containers can be added so the transition to* $X_{n+1} is based on the probability that all$ *none of the containers being removed* (0.2^3) *and 1 or 2 containers being removed* ($3(.8)(.2^2)$ *and* $3(.8)^2(.2)$)*.*

(b) Find the probability that there are 3 containers in the stockyard at the end of day 3, given that there were 2 containers there at the end of day 1.

To obtain this probability, we need to evaluate the 3-4 entry of the two step transition matrix:

$$P(X_3 = 3|X_1 = 2) = P_{34}^{(2)} = .096(.008) + .008(.008) = 0.000832.$$

(c) Is the state space irreducible? Explain. *The state space is irreducible since all states communicate (i.e. the Markov chain can transit from 0 to 1 to 2 to 3 and back to 0 with nonzero probability.*

(d) If there is a limiting distribution, find it.

There is a limiting stationary distribution, because the Markov chain is aperiodic and irreducible.

Solving $\pi =$ *with* $\pi_0 + \pi_1 + \pi_2 + \pi_3 = 1$*, any way that you wish, gives*

$$\pi^\top = [0.7603 \quad 0.2294 \quad 0.0102 \quad 0.0001].$$

5. Consider the time-reversible Markov chain discussed in class:

$$P_{i,j} = \frac{1}{6}\min\left(\frac{\pi_j}{\pi_i}, 1\right), \text{ for } j = i-2, i-1, i+1, i+2$$

and 0 for $|j - i| > 2$. $P_{i,i}$ is set to ensure that the row sums of P are 1.

Use this in an MCMC simulation of the probability distribution $\pi_j = P(X = j) = k(.7)^{j-1}$ for $j = 1, 2, \ldots$, where k is a value that could be calculated but is not needed. Simulate 20000 values and use these to estimate the probability π_3.

We modify the function that computes the unnormalized values of the steady-state (target) distribution as follows:

```
pi.fun <- function(i) {
    out <- 0
    if (i > 0) out <- .7^(i-1)
    out
}
```

No modification of the Metropolis-Hastings code from class is needed:

```
Ntransitions <- 20000
X <- numeric(Ntransitions)
current.state <- 50  # initialize the Markov chain
for (n in 1:Ntransitions) {
     i <- current.state
     P <- c(min(pi.fun(i-2)/pi.fun(i), 1),
     min(pi.fun(i-1)/pi.fun(i), 1),
     min(pi.fun(i+1)/pi.fun(i), 1),
     min(pi.fun(i+2)/pi.fun(i), 1))/6
```

```
    P0 <- 1 - sum(P)
    P <- c(P[1:2], P0, P[3:4])
    transition <- sample(seq(-2,2,1), size = 1, prob = P)
    current.state <- current.state + transition
    X[n] <- current.state
}
observedDist <- table(X[-c(1:1000)])
```

The estimated probability, π_3 *is* $2882/19000 = 0.152$*. (Incidentally, the true value is* $.3(.7)^2 = .147$*.)*