Stats 207: HW2

Problem 1

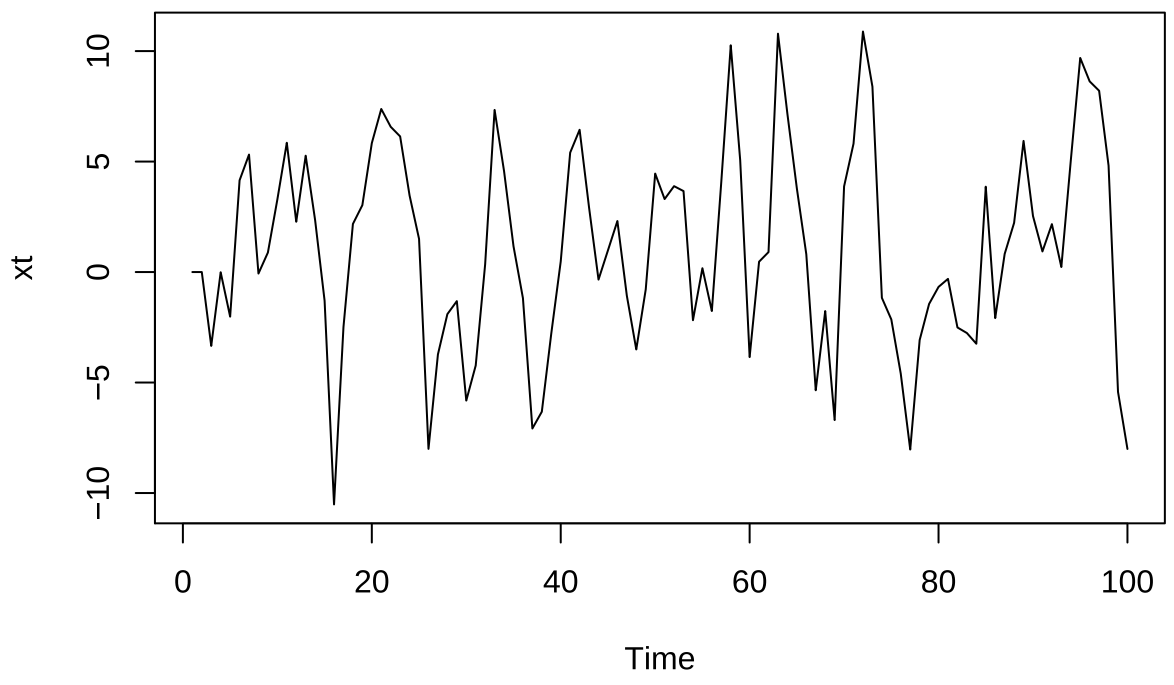
$Z_t \overset{iid}{\sim} N(0,1)$ and $C_t \overset{iid}{\sim} \text{Cauchy}(0,1)$

Part a

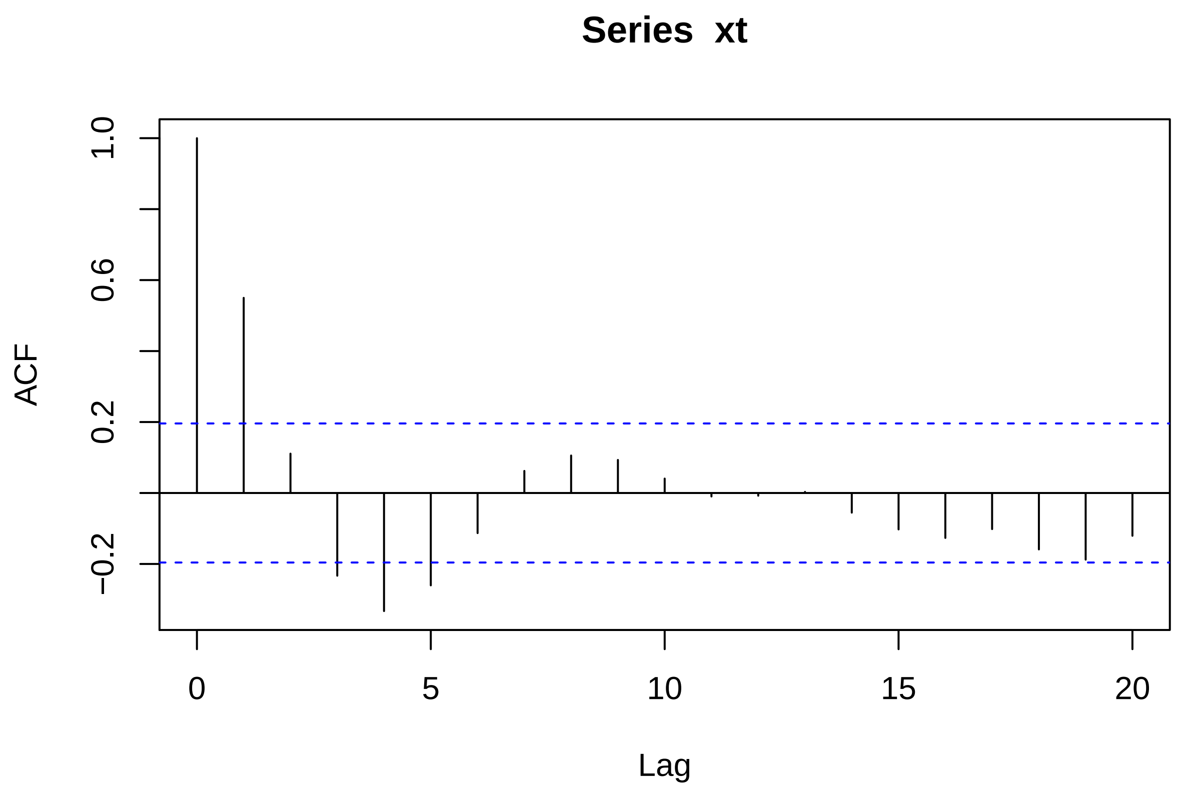
For $t \in [0, 100]$, simulate ARMA(2,2):

$$X_t = 0.7X_{t-1} - 0.3X_{t-2} + Z_t + 4Z_{t-2}$$

```r
set.seed(1)
t = 100
zt = rnorm(t, 0, 1)
xt = c(0,0)

for (t in (3:t)){
  xt[t] = 0.7 * xt[t-1] - 0.3 * xt[t-2] + zt[t] + 4* zt[t-2]
}

# arima.sim(model = list(order = c(2, 0, 1), ar = c(0.7, -0.3), ma = c(4)), n = t, n.start = 3, innov =
plot(xt, type = 'l', xlab = "Time")
```



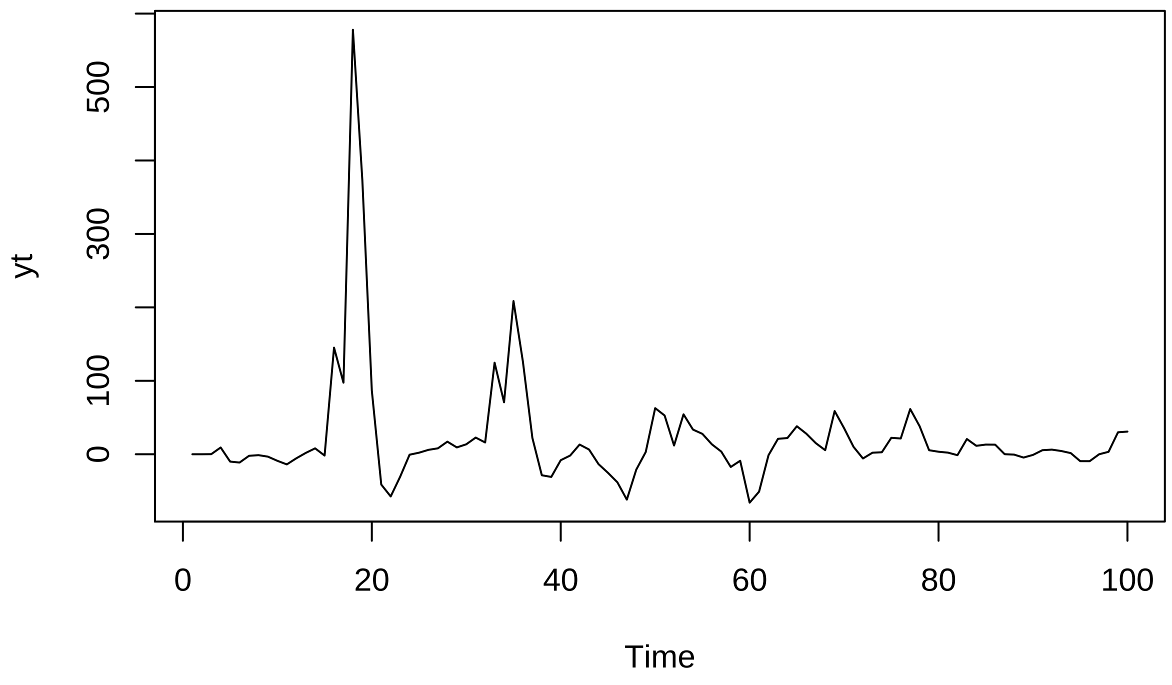
```r
acf(xt)
```



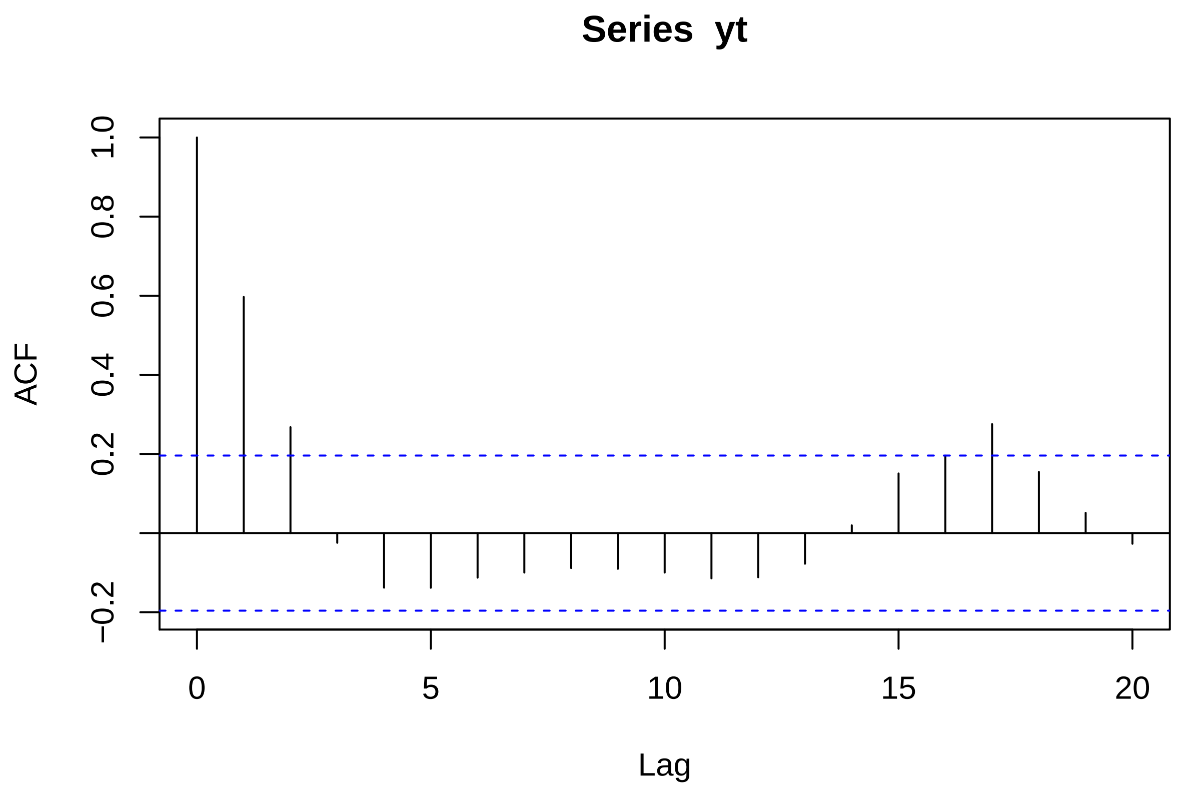
Part b

$$Y_t = 0.7Y_{t-1} - 0.3Y_{t-2} + C_t + 4C_t$$

```
set.seed(1)
ct = rcauchy(t, 0, 1)
yt = c(0,0)

for (t in (3:t)){
  yt[t] = 0.7 * yt[t-1] - 0.3 * yt[t-2] + ct[t] + 4* ct[t-2]
}

plot(yt, type = 'l', xlab = "Time")
```



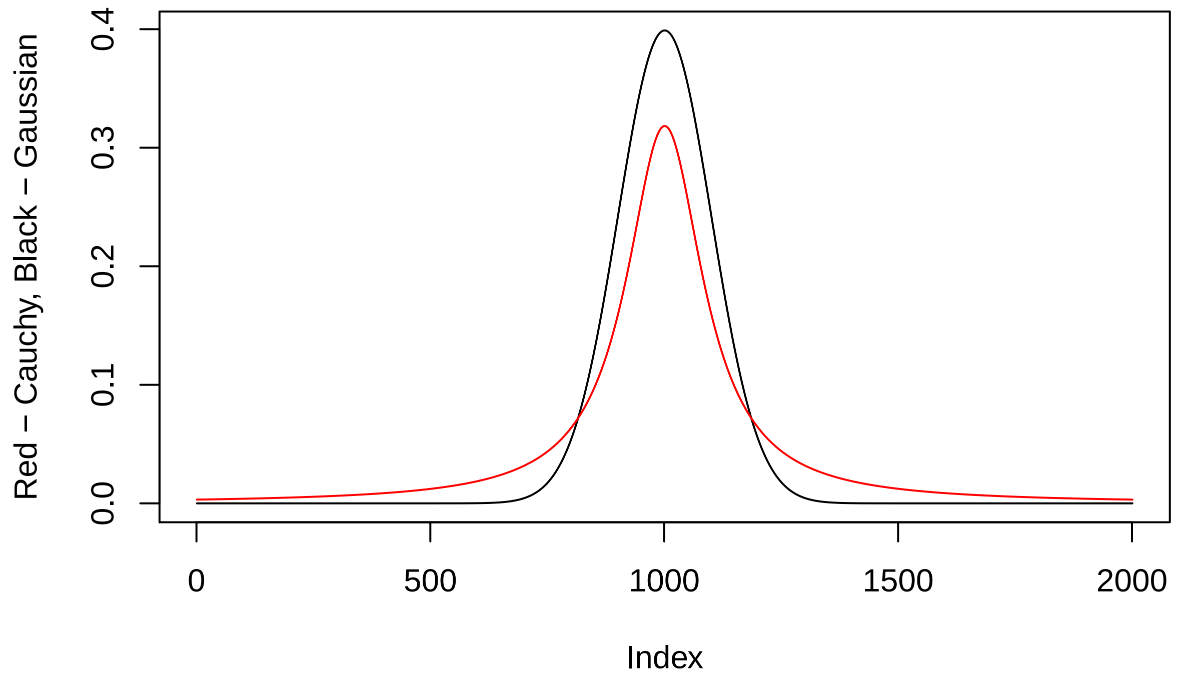
```r
acf(yt)
```



Part c

The Cauchy ARMA process differs from the Gaussian by having extreme observations occur randomly throughout the sequence. This is because the Cauchy distribution has a much longer tail, which makes the occurence of an extreme value much more likely. And because the ARMA process is recursively defined, the effects of these extreme observations continue to influence future observations. Accordingly, the ACF of the Cauchy ARMA process exhibits greater correlation since these extreme observations are correlating with themselves from the past. Besides from these extreme values, the sequences appear to behave somewhat similarly.

```r
plot(dnorm(seq(-10,10,.01)), type = "l", ylab = "Red - Cauchy, Black - Gaussian")
lines(dcauchy(seq(-10,10,.01)), type = "l", col = "red")
```



Part D

Plot the theoretical ACF with a 95% confidence interval. The confidence interval is given by:

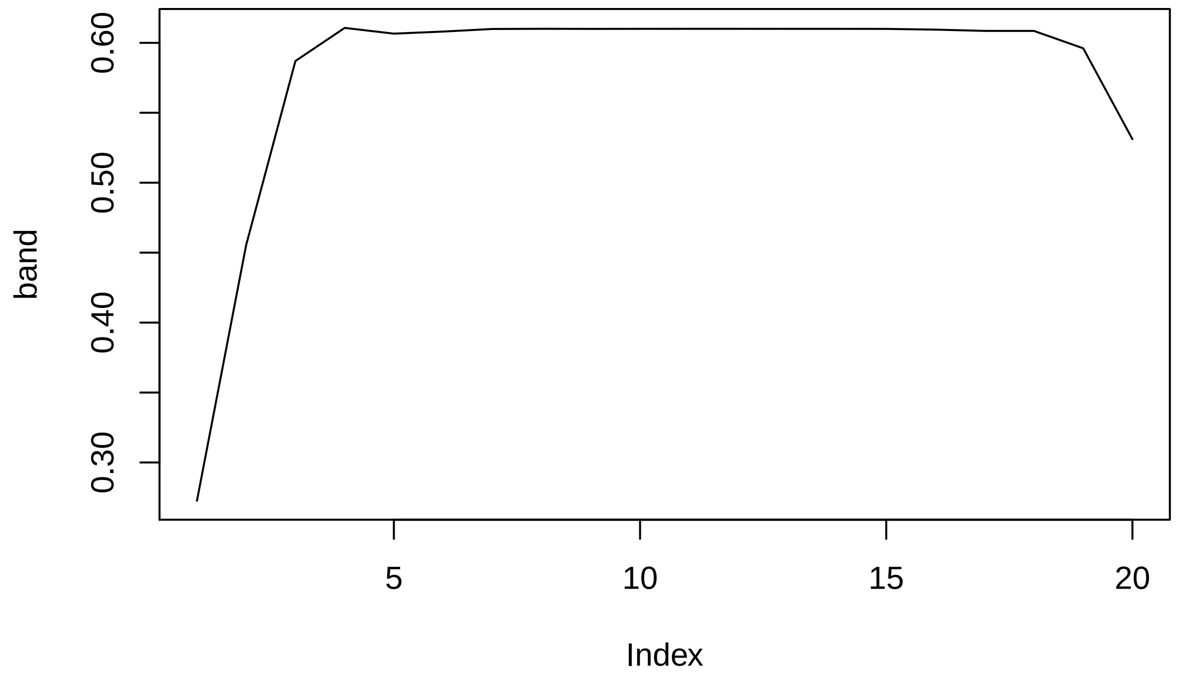
$$\rho_X(i) \pm 1.96\sqrt{W_{ii}/n}$$

Where,

$$W_{ii} = \sum_{\infty} (\rho(m+i) + \rho(m-i) - 2\rho(i)\rho(m))^2$$

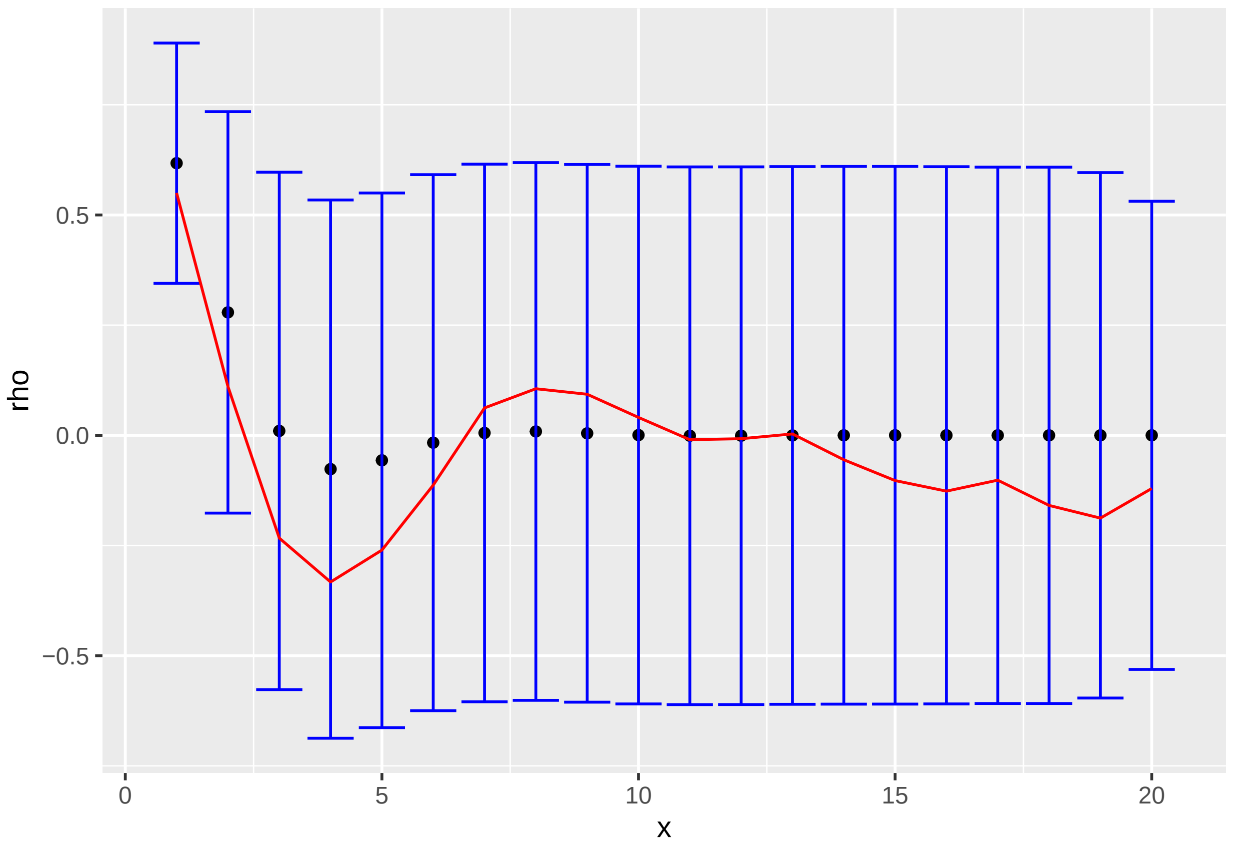
We will use a summation over the first 20 lags as an approximation.

```r
n = 20
rho = ARMAacf(ar = c(0.7,-0.3), ma = c(0,4), lag.max = 40)
Wii = rep(0, n)

for(i in 1:n){
  for(m in 1:n){
      k = 0
      k = (rho[(m + i)+1] + rho[abs(m - i)+1] - 2*rho[i+1]*rho[m+1])^2
      Wii[i] = Wii[i] + k
  }
}

band = 1.96 * sqrt(Wii / n)
plot(band, type = 'l')
```



In this plot, we can see that the error bands at first grow low to high, then stabilize, until approaching a lag of $h = 20$ where it drops off sharply.

```r
ggplot(data.frame(rho = rho[2:(n+1)], band, x = 1:n, sample = acf(xt,plot = FALSE, lag.max = n)[["acf"]
  geom_point() +
  geom_errorbar(aes(ymin = rho - band, ymax = rho + band), col = "blue") +
  geom_line(aes(y = sample), col = "red")
```



Here we see the theoretical ACF with 95% confidence error bands plotted with the sample ACF from our simulation. We see that the realization fluctuates around the theoretical values but never differs by what is suggested with 95% confidence. In fact, the series appears to match the theoretical values for moderate lag size before trailing off around $h = 20$. This is evidence that simulation and theoretical values are indeed from the same model.

Using the fact that,

$$X_t = \psi(B)Z_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}$$

Thus,

$$X_0 = \sum_{j=0}^{20} \psi_j Z_{0-j}$$

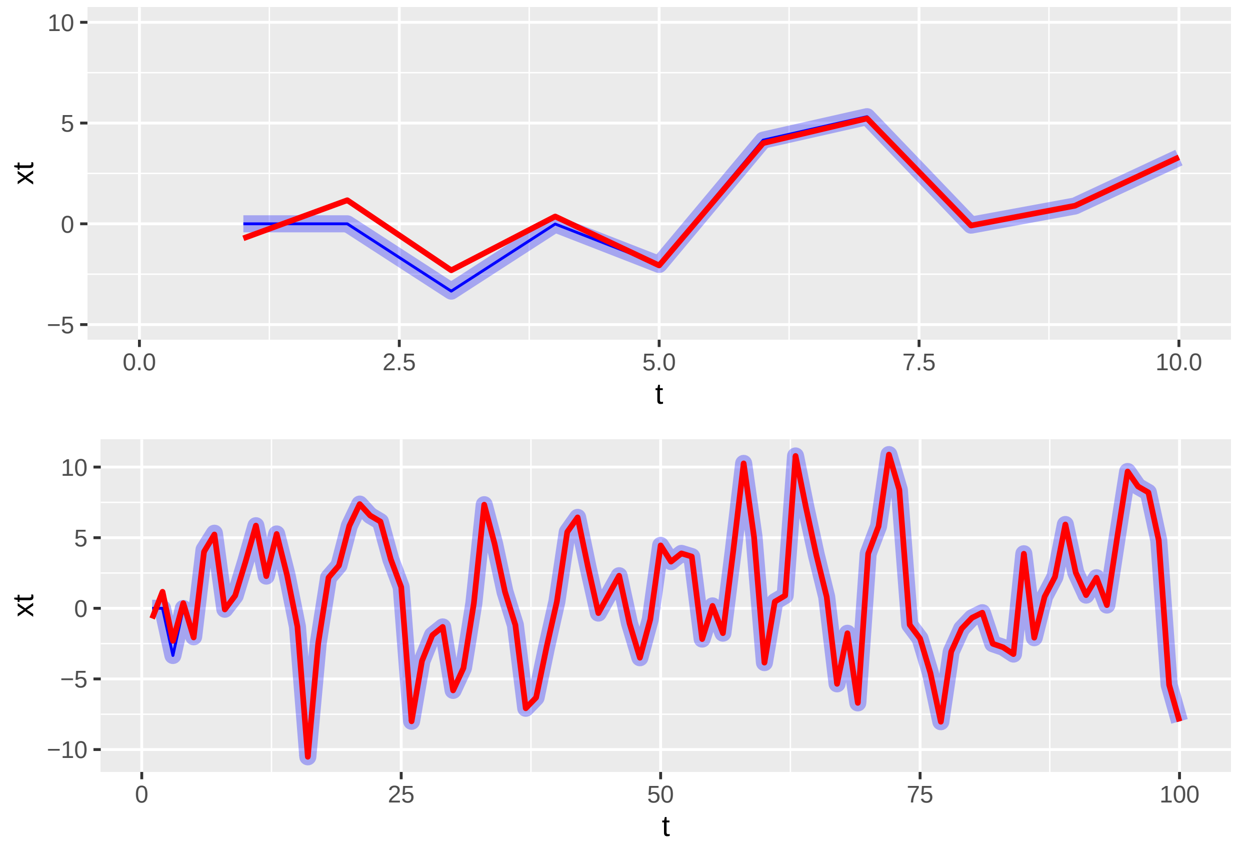
and,

$$X_1 = \sum_{j=0}^{20} \psi_j Z_{1-j}$$

We can compute X_0, X_1 using the MA(∞) representation:

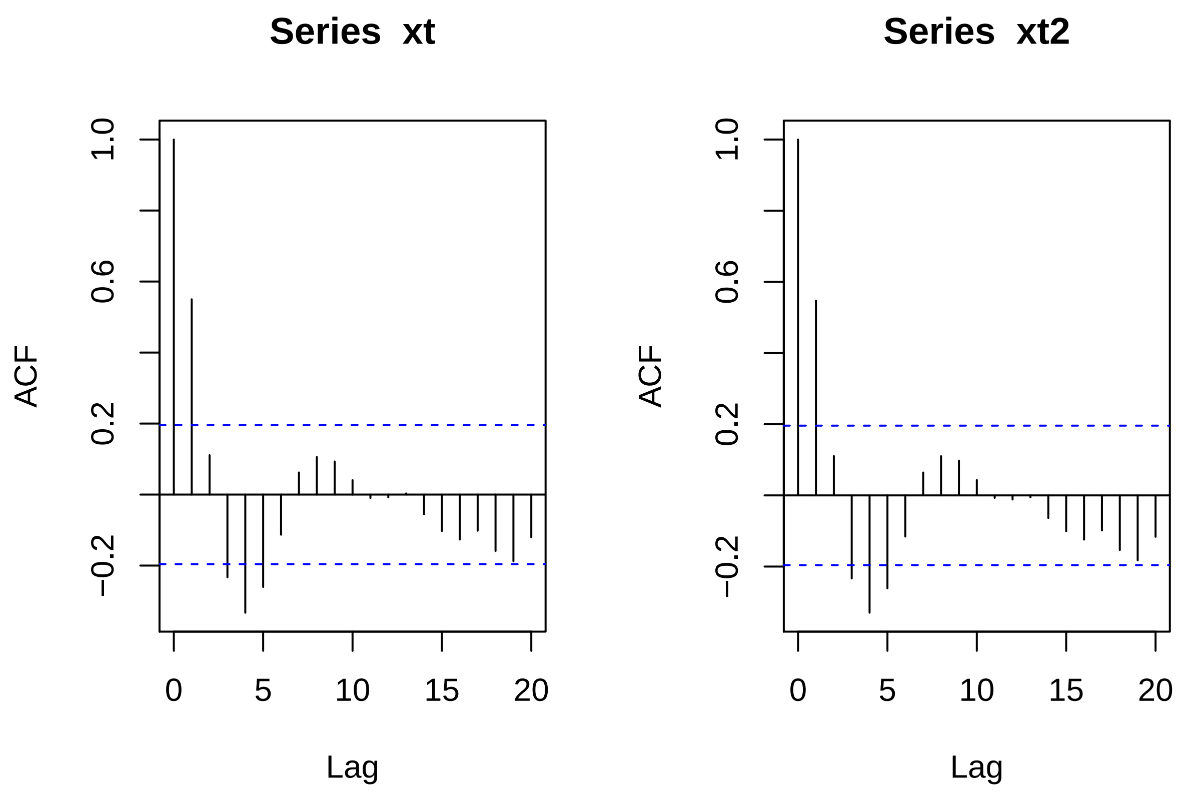
```r
set.seed(1)
psi = ARMAtoMA(ar = c(0.7,-0.3), ma = c(0,4), lag.max = 20)
zt2 = rnorm(19, 0, 1)
xt2 = c()
xt2[1] = c(zt2,zt[1]) %*% psi
# Combine the zt2 with the first observation of zt for X_1
xt2[2] = c(zt2[2:19], zt[1], zt[2]) %*% psi
```

```r
for (t in (3:t)){
  xt2[t] = 0.7 * xt2[t-1] - 0.3 * xt2[t-2] + zt[t] + 4* zt[t-2]
}

p1 = ggplot(data.frame(xt, xt2, t = 1:t), aes(x = t)) +
  geom_line(aes(y = xt), col = 'blue', size = 3, alpha = .3) +
  geom_line(aes(y = xt), col = 'blue') +
  geom_line(aes(y = xt2), col = 'red', size = 1) + xlim(0, 10) + ylim(-5,10)
p2= ggplot(data.frame(xt, xt2, t = 1:t), aes(x = t)) +
  geom_line(aes(y = xt), col = 'blue', size = 3, alpha = .3) +
  geom_line(aes(y = xt), col = 'blue') +
  geom_line(aes(y = xt2), col = 'red', size = 1)
grid.arrange(p1, p2,ncol = 1)
```



Plotting this new simulation's data with that of the original simulations, we see that they are different up to $t = 5$ and then afterwards are identical. This stands to reason from the ψ weights:
ψ : 0.7, 4.19, 2.723, 0.6491, -0.36253, -0.448501, -0.2051917, -0.0090839, 0.0551988, 0.0413643, 0.0123954, -0.0037325, -0.0063314, -0.0033122, $-4.1913261 \times 10^{-4}$, 7.0027043×10^{-4}, 6.1592909×10^{-4}, 2.2106923×10^{-4}, $-3.0030265 \times 10^{-5}$, $-8.7341955 \times 10^{-5}$
Become smaller and smaller with larger lags, thus resulting in identical process uninfluenced from beginning observations for a sufficiently large lag.

```r
par(mfrow = c(1,2))
acf(xt)
acf(xt2)
```



The sample ACFs look basically the same since the majority of the two simulations are identical.

Part f

To solve for the variance of the ARMA(2, 2),

$$X_t = 0.7X_{t-1} - 0.3X_{t-2} + Z_t + 4Z_{t-2}$$

We have to set up a linear system of $p+1$ equations to solve for $p+1$ unknowns, $\gamma(0), ..., \gamma(p)$. Recall,

$$\gamma_X(k) - \phi_1\gamma_X(k-1) - ... - \phi_p\gamma_X(k-p) = c_k, \forall k \geq 0$$

Where,

$$c_k = (\psi_0\theta_k + \psi_1\theta_{k+1} + ... + \psi_{q-k}\theta_q)\sigma_Z^2 : 0 \leq k \leq q, 0 : k > q$$

Thus with $p = 2, k = 2$,

$$\gamma_X(0) - \phi_1\gamma_X(-1) - \phi_2\gamma_X(-2) = (\psi_0\theta_0 + \psi_1\theta_1 + \psi_2\theta_2)\sigma_Z^2$$

$$\gamma_X(1) - \phi_1\gamma_X(0) - \phi_2\gamma_X(-1) = (\psi_0\theta_1 + \psi_1\theta_2)\sigma_Z^2$$

$$\gamma_X(2) - \phi_1\gamma_X(1) - \phi_2\gamma_X(0) = \psi_0\theta_2\sigma_Z^2$$

Noting that $\gamma(k) = \gamma(-k)$, we can then solve for this in matrix algebra,

$$\begin{bmatrix}\gamma(0) & \gamma(1) & \gamma(2)\end{bmatrix} \cdot \begin{bmatrix}1 & -\phi_1 & -\phi_2 \\ -\phi_1 & 1-\phi_2 & -\phi_1 \\ -\phi_2 & 0 & 1\end{bmatrix} = \begin{bmatrix}(\psi_0\theta_0 + \psi_1\theta_1 + \psi_2\theta_2)\sigma_Z^2 \\ (\psi_0\theta_1 + \psi_1\theta_2)\sigma_Z^2 \\ \psi_0\theta_2\sigma_Z^2\end{bmatrix}$$

Plugging in $(\theta_1 = 0, \theta_2 = 4, \phi_1 = 0.7, \phi_2 = -0.3)$ values gives,

```
psi
```

```
##  [1]  7.000000e-01  4.190000e+00  2.723000e+00  6.491000e-01 -3.625300e-01
##  [6] -4.485010e-01 -2.051917e-01 -9.083890e-03  5.519879e-02  4.136432e-02
## [11]  1.239539e-02 -3.732525e-03 -6.331383e-03 -3.312211e-03 -4.191326e-04
## [16]  7.002704e-04  6.159291e-04  2.210692e-04 -3.003026e-05 -8.734195e-05
```

$$\begin{bmatrix}\gamma(0) & \gamma(1) & \gamma(2)\end{bmatrix} \cdot \begin{bmatrix}1 & -0.7 & 0.3 \\ -0.7 & 1.3 & -0.7 \\ 0.3 & 0 & 1\end{bmatrix} = \begin{bmatrix}17.76 & 2.8 & 4\end{bmatrix}$$

```
gamma = as.vector(t(c(1 + psi[2]*4, psi[1]*4, 1*4)) %*% solve(rbind(c(1,-0.7, 0.3), c(-0.7, 1.3, -0.7),
gamma
```

```
## [1] 27.261905 16.833333  7.604762
```

Now using the difference equation,

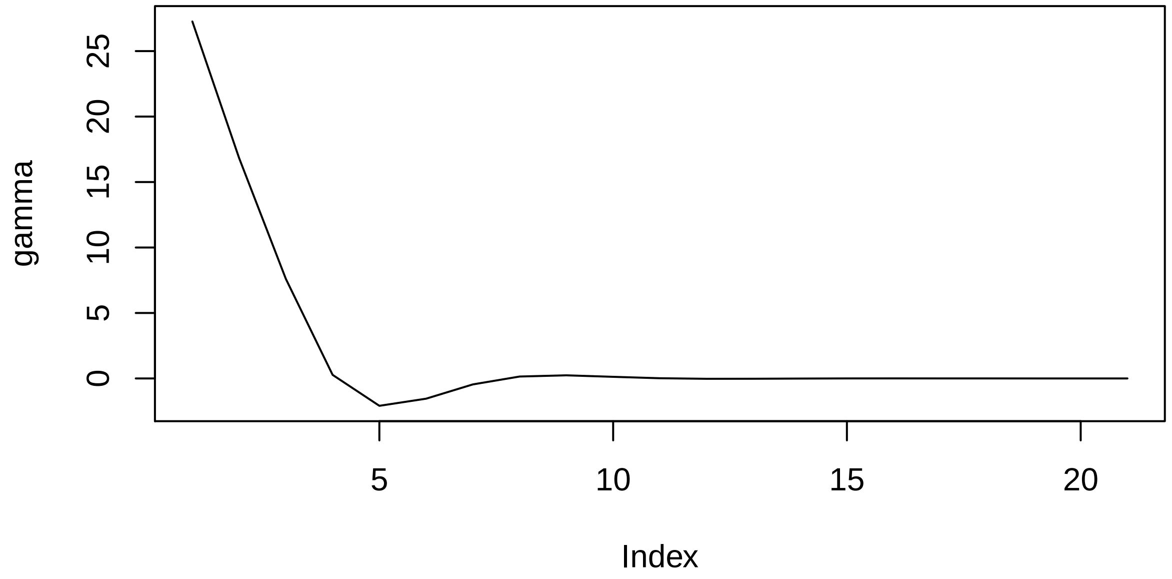
$$\gamma_X(k) = c_k + \phi_1\gamma_X(k-1) + ... + \phi_p\gamma_X(k-p), k > p$$

$$\gamma_X(4) = \phi_1\gamma_X(3) + ... + \phi_2\gamma_X(2), k > p$$

$$\vdots$$

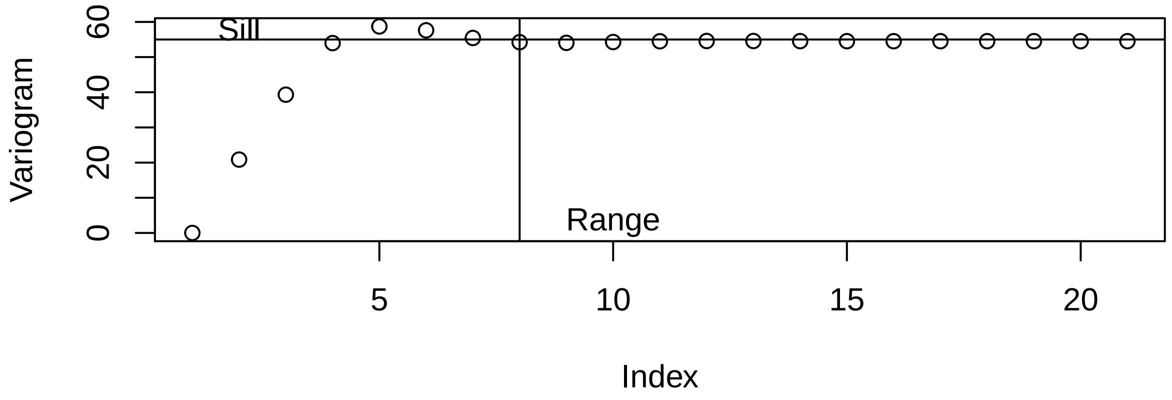
$$\gamma_X(20) = \phi_1\gamma_X(19) + ... + \phi_2\gamma_X(18), k > p$$

```r
for(i in 4:21){
  gamma[i] = 0.7*gamma[i-1] - 0.3*gamma[i-2]
}
plot(gamma, type = "l")
```



Here we see a plot of the autocovariance function. Next is a bonus plot of the variogram, $2\gamma = 2[C(0) - C(h)]$ with no nugget effect:

```r
plot(2*(gamma[1] - gamma), ylab = "Variogram"); abline(h = 55); abline(v = 8); text(2,58, "Sill"); text
```

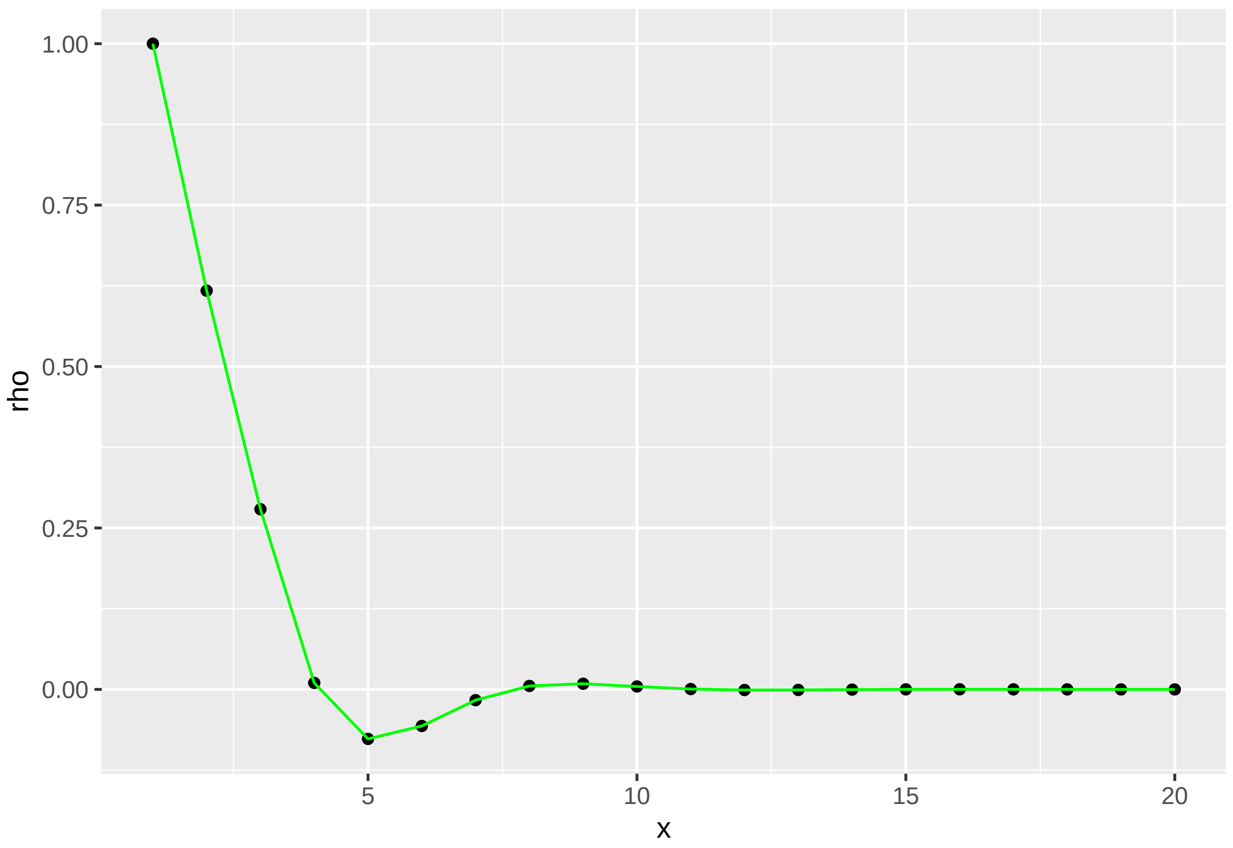


Part g

```
rho_t = gamma / gamma[1]
rho_t
```

```
##  [1]  1.000000e+00  6.174672e-01  2.789520e-01  1.002620e-02 -7.666725e-02
##  [6] -5.667493e-02 -1.667228e-02  5.331885e-03  8.734003e-03  4.514237e-03
## [11]  5.397648e-04 -9.764357e-04 -8.454344e-04 -2.988734e-04  4.441895e-05
## [16]  1.207553e-04  7.120301e-05  1.361552e-05 -1.183004e-05 -1.236568e-05
## [21] -5.106967e-06
```

```
ggplot(data.frame(rho = rho[1:n], rho_t = rho_t[1:n], band, x = 1:n, sample = acf(xt,plot = FALSE, lag.
  geom_point() +
  geom_line(aes(y = rho_t), col = "green")
```



It appears that the theoretical values from the differencing equation match that of `ARMAacf()` exactly.

Theory 3

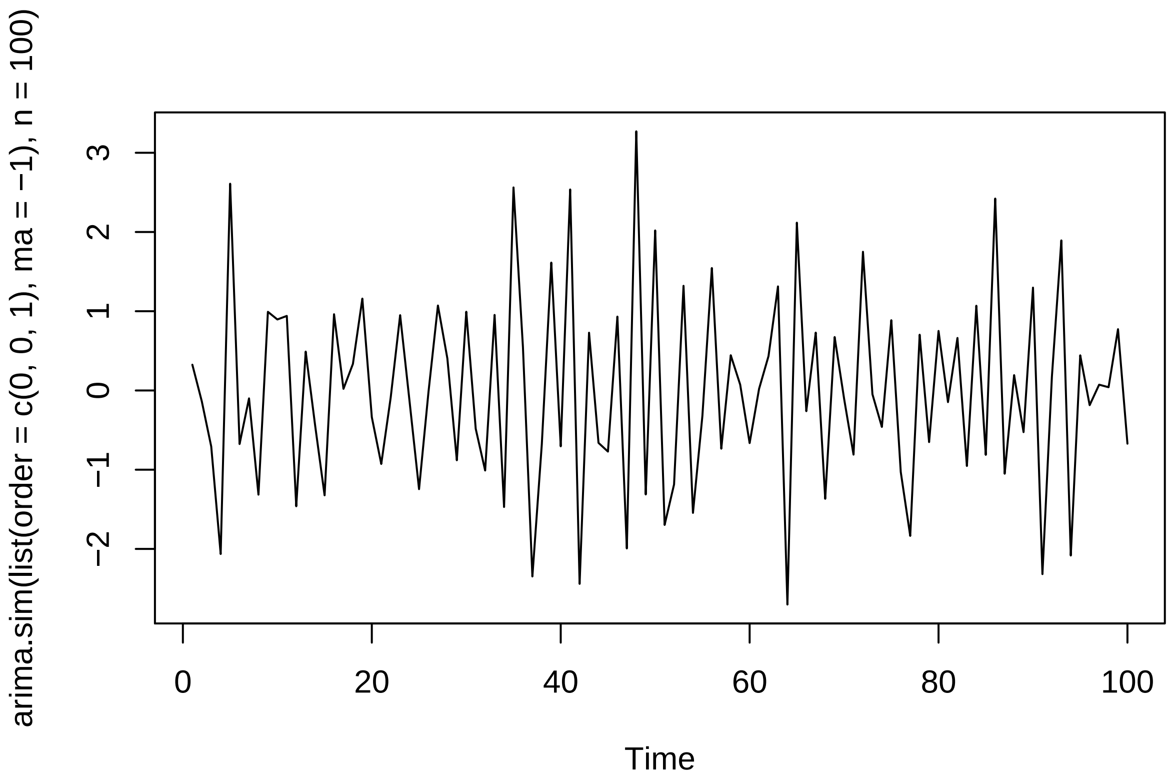
```r
ARMAtoMA(ar = .5, ma = c(0,1),lag.max = 5)
```

```
## [1] 0.50000 1.25000 0.62500 0.31250 0.15625
```

```r
ARMAacf(ar = .5, ma = c(0,1), lag.max = 5)
```

```
##        0        1        2        3        4        5
## 1.000000 0.650000 0.625000 0.312500 0.156250 0.078125
```

```r
plot(arima.sim(list(order = c(0,0,1), ma = -1), n = 100))
```


arima.sim(list(order = c(0, 0, 1), ma = −1), n = 100)
Time