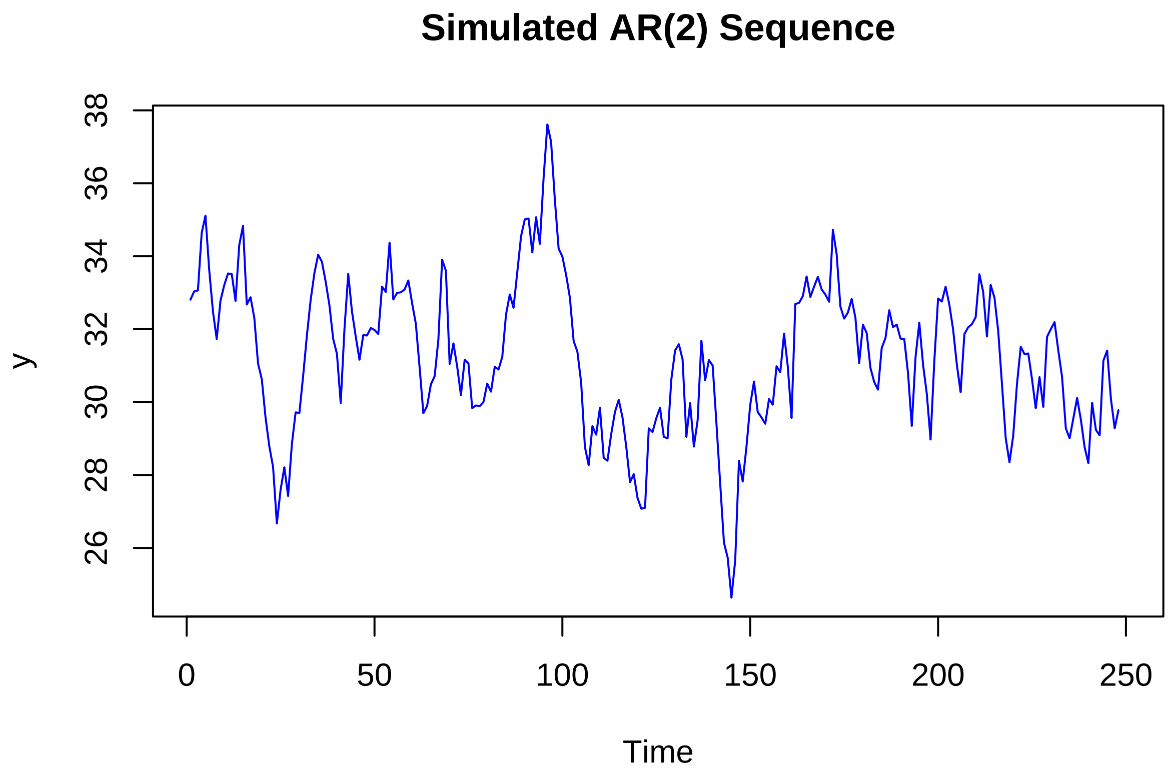
HW 5

2023-05-15

1)

```r
# Set parameters
n = 250

# Set seed
set.seed(123)

# Generate white noise. e ~ N(0,1)
ep = rnorm(n, 0, 1)

# Generate the AR(2) sequence
y = rep(NA, n+2)
y = as.ts(y)
y[1] = 31.25 # Use fixed point as starting seed
y[2] = 31.25

# Simulate sequene
for (sim in 3:n){
  y[sim] = 2.5 + 1.1*y[sim-1] - 0.18*y[sim-2] + ep[sim]
}

# Get rid of starting seed
y = y[-c(1,2)] |> as.ts()

# Plot the sequence
plot(y, col = 'blue', main = 'Simulated AR(2) Sequence')
```



It looks covariance stationary since the sequence stays around fairly steady, and doesn't expload in value.

2)

```r
# Vector of coefficients for AR(2) sequence
poly_y = c(1, -1.1, 0.18)

# Calculate the roots
(roots = polyroot(poly_y))
```

```
## [1] 1.111111+0i 5.000000-0i
```

```r
# Companion F matrix
f_matrix = matrix(c(1.1, -.18, 1, 0), nrow = 2, ncol = 2, byrow = TRUE)

# Calculate eigenvalues
(eigen_values = eigen(f_matrix)$value)
```

```
## [1] 0.9 0.2
```

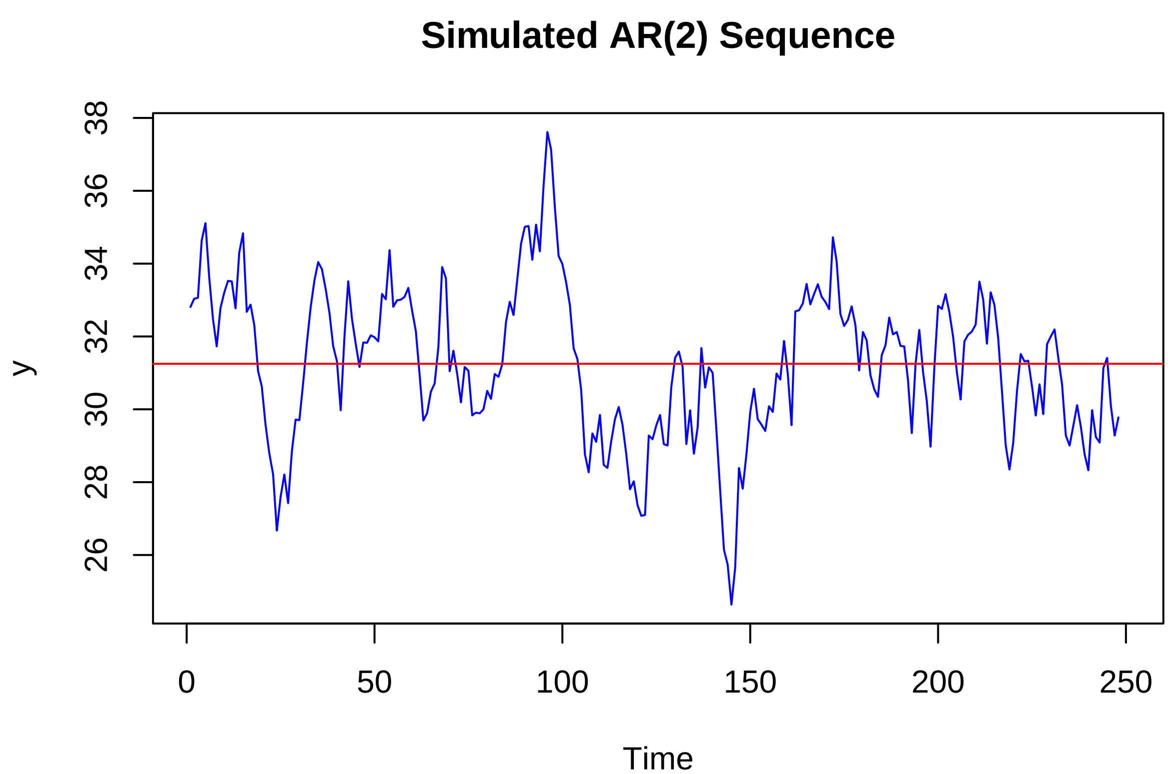
Both roots are outside the complex unit circle, 1.1111111 and 5, so the processs is stationary.

3)

```
# Calculate the unconditional mean
(mean_y = 2.5/(1 -1.1 +0.18))
```

```
## [1] 31.25
```

```
# Put mean on graph
plot(y, col = 'blue', main = 'Simulated AR(2) Sequence')
abline(h = mean_y, col = 'red')
```

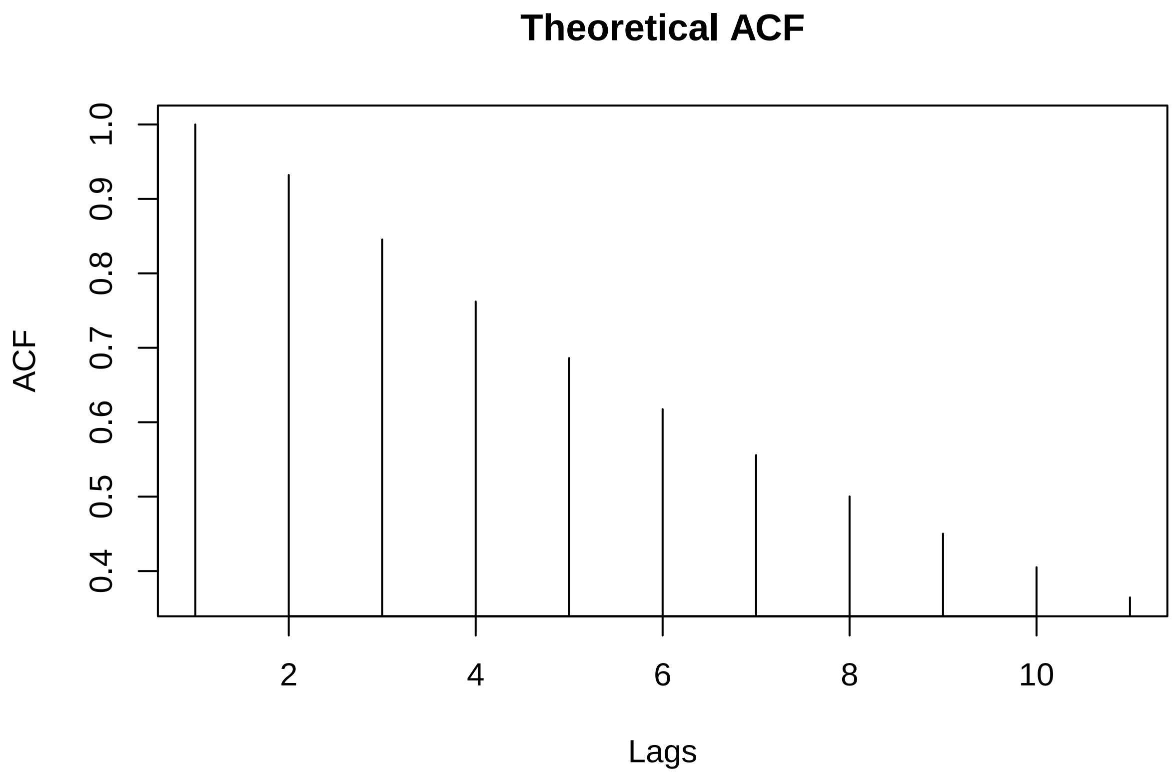


4)

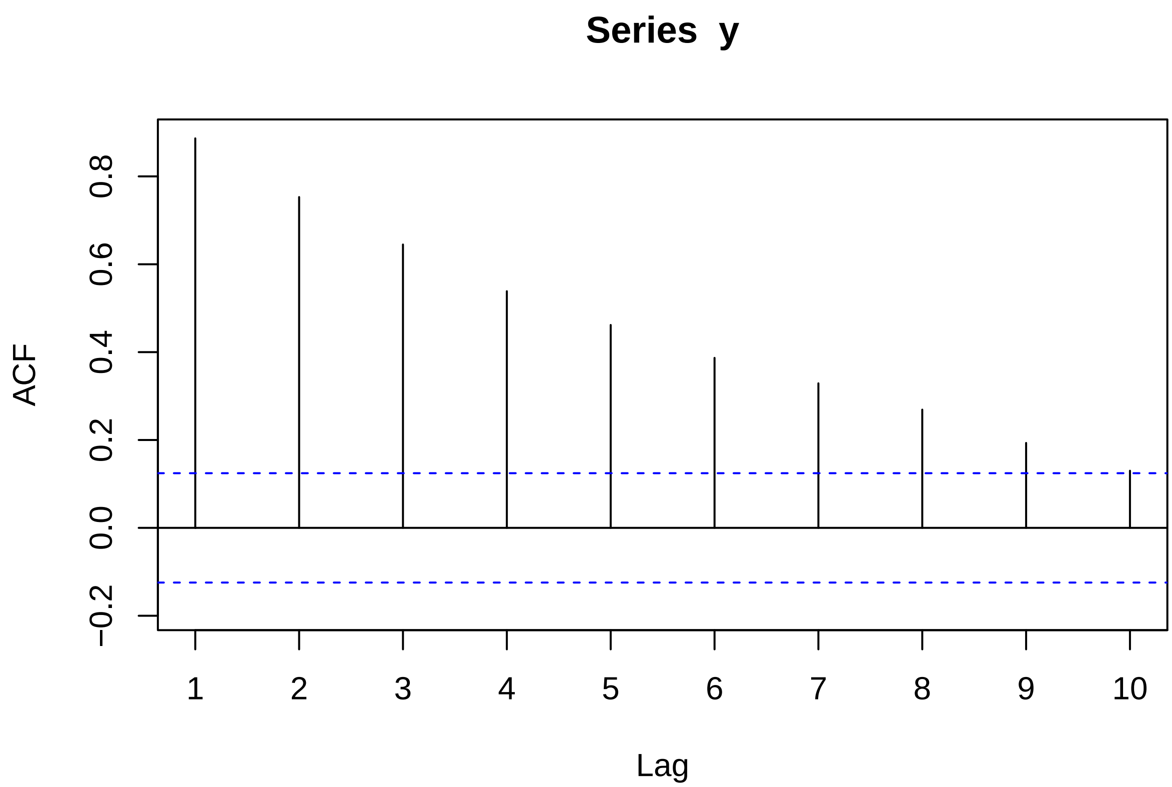
```
# Calculate theorertical autocorrelations
(theo_acf = ARMAacf(ar = c(1.1, -0.18), ma = 0, lag.max = 10))
```

```
##         0         1         2         3         4         5         6         7 
## 1.0000000 0.9322034 0.8454237 0.7621695 0.6862102 0.6176407 0.5558869 0.5003003 
##         8         9        10 
## 0.4502707 0.4052437 0.3647193
```

```r
# Plot them
plot(theo_acf, type = 'h', main = 'Theoretical ACF', ylab = 'ACF', xlab = 'Lags')
```




```r
# Plot sample ACF
Acf(y, lag.max = 10)
```

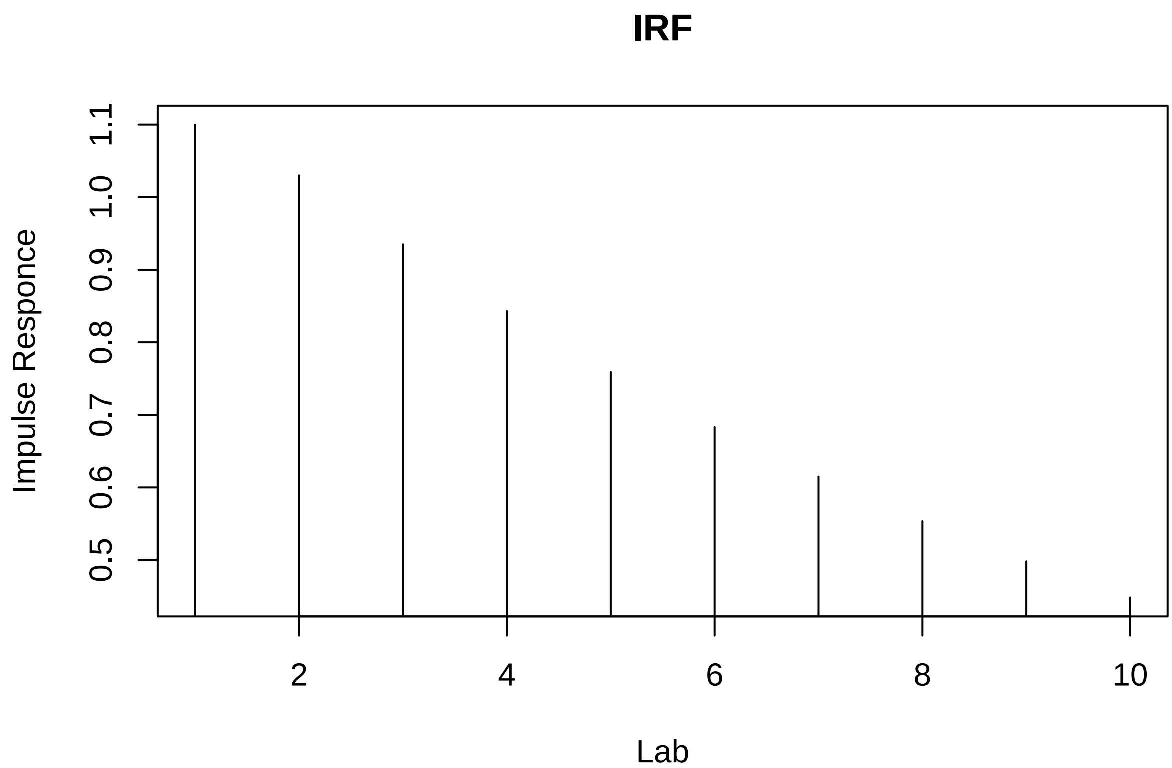


5)

```
# Convert to moving average
(irf_y = ARMAtoMA(ar = c(1.1, -0.18), ma = 0, lag.max = 10))
```

```
##  [1] 1.1000000 1.0300000 0.9350000 0.8431000 0.7591100 0.6832630 0.6149495
##  [8] 0.5534571 0.4981119 0.4483008
```

```
# Plot IRF
plot(irf_y, type = 'h', main = 'IRF', ylab = 'Impulse Responce', xlab = 'Lab')
```



The IRF is very similar to the ACF, but the differ by a bit.