HOMEWORK ASSIGNMENT 6

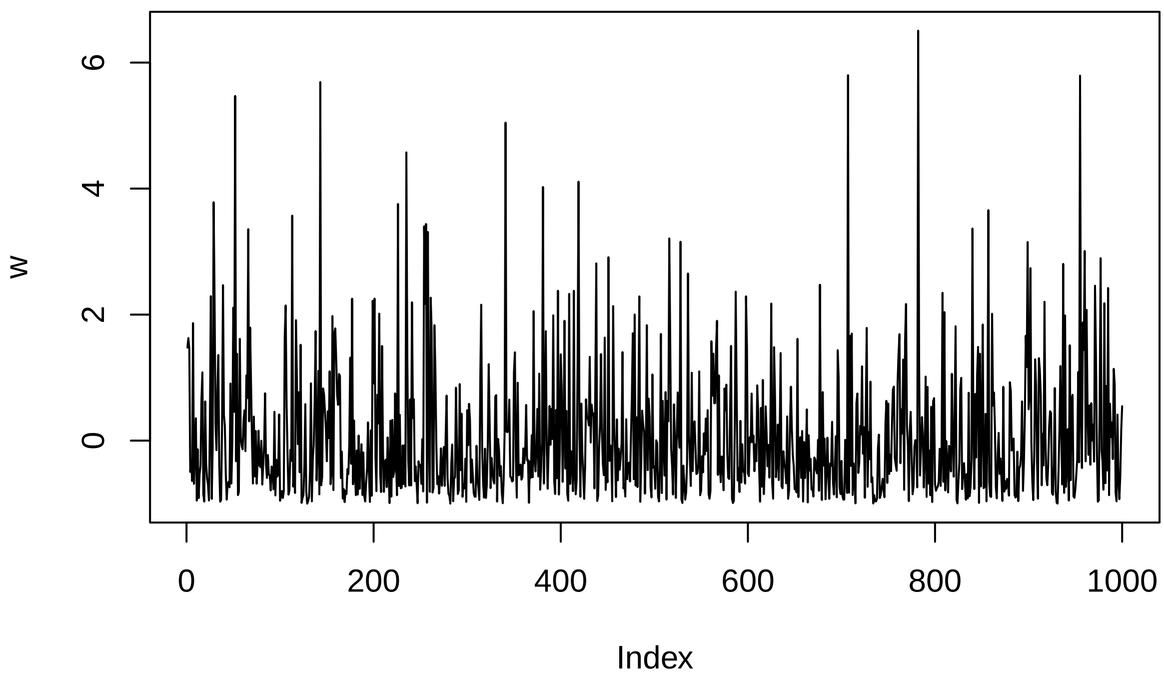
Zachary Lazerick

12 March 2023

PROBLEM 1: Exponential White Noise

(a) Simulate discrete white noise from an exponential distribution and plot the data using the `plot()` function. You can use the `R` command `w <- rexp(1000)-1` for exponential white noise with zero mean.

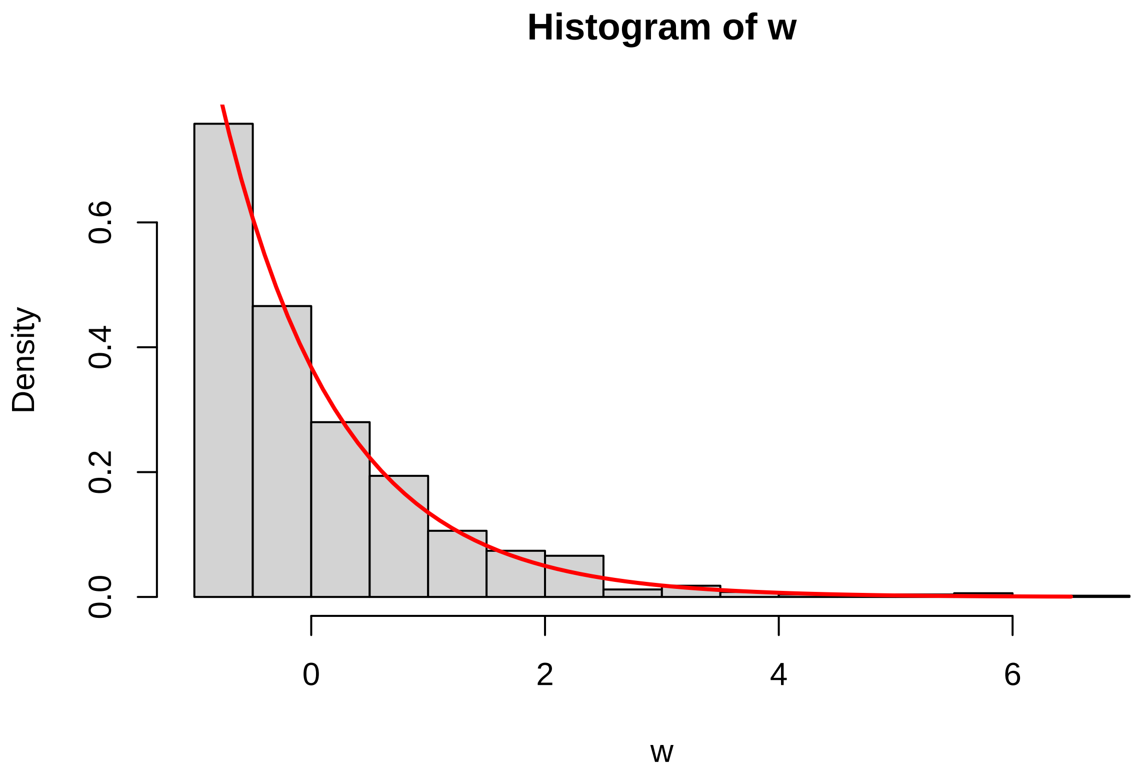
```
set.seed(721)
w <- rexp(1000) - 1

plot(w, type = 'l')
```



(b) Plot the histogram and the underlying true distribution using `hist()`, `points()`, and `dexp()`. Does the true distribution correctly model the data?

```
hist(w, freq = F)
x <- seq(-1, max(w), .1); dexp <- dexp(x+1)
```

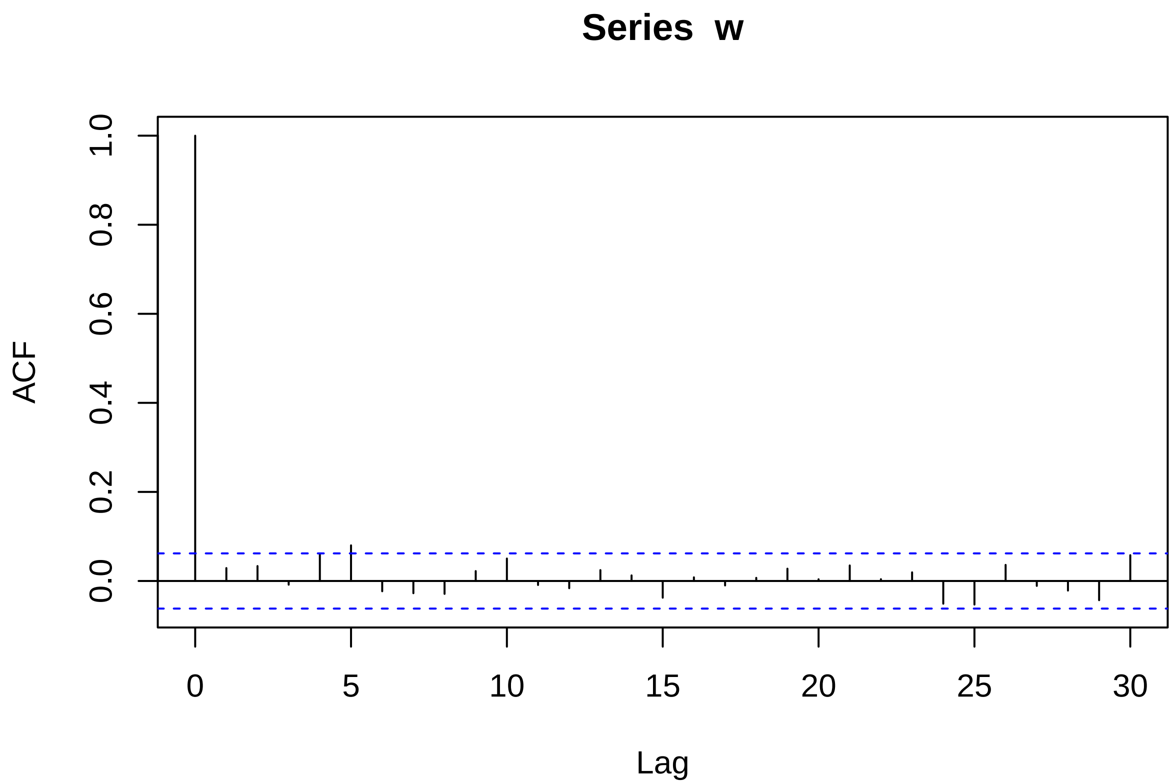
```r
lines(x, dexp, col = 'red', lwd = 2)
```



The true distribution does an acceptable job at modeling the data. The deviance between the true can be attributed to the random generation of the white noise.

(c) Plot the autocorrelation function using `acf()`. Does this plot confirm that you indeed simulated white noise? Explain.

```r
acf(w)
```



The acf plot confirms that we simulated white noise because the autocorrelation is within the confidence band for all lags greater than zero. This means that the correlation between points at all levels of lag is likely insignificant.

PROBLEM 2: AR(1) Models

(a) Simulate time series of length 100 from an AR(1) model with α equal to -0.8. Then estimate the parameter and order of the model using the `ar()` function.

```
set.seed(831)
x <- w <- rnorm(100); alpha <- -0.8
for (i in 2:length(x)) {
  x[i] <- alpha*x[i-1] + w[i]
}

x.ar <- ar(x)

## Parameter; Order; Estimates
x.ar$ar; x.ar$order
```

```
## [1] -0.9148065

## [1] 1
```

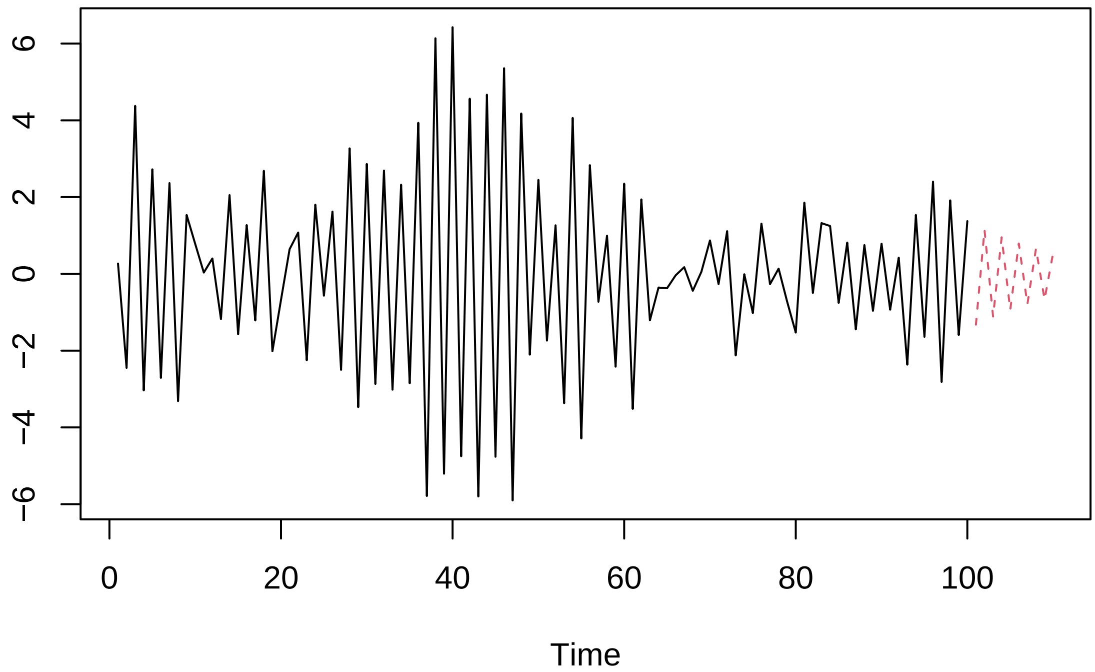
(b) Generate a 95% confidence interval for α:

```
x.ar$ar + c(-1.96, 1.96) * sqrt(x.ar$asy.var.coef[1, 1])
```

```
## [1] -0.9947731 -0.8348399
```

(c) Make predictions for 1 to 10 steps ahead using the `predict()` function and plot the function along with the predictions using `ts.plot()`. Remember one can set different line styles with `lty=1:2` and different colors with `col=1:2`.

```
set.seed(807)
x.ts <- ts(x); x.predict <- predict(x.ar, n.ahead = 1 * 10)

ts.plot(x.ts, x.predict$pred, col = 1:2, lty = 1:2)
```



d) Simulate time series of length 100 from an AR(1) model with α equal to -0.4. Then estimate the parameter and order of the model using the `ar()` function. Make predictions for 1 to 10 steps ahead using the `predict()` function and plot the function along with the predictions using `ts.plot()`. How do these predictions compare with those from part c)?

```
set.seed(811)
x <- w <- rnorm(100); alpha <- -0.4
for (i in 2:length(x)) {
  x[i] <- alpha*x[i-1] + w[i]
}

x.ar <- ar(x)

## Parameter; Order; Estimates
x.ar$ar; x.ar$order
```

```
## [1] -0.2276263
```

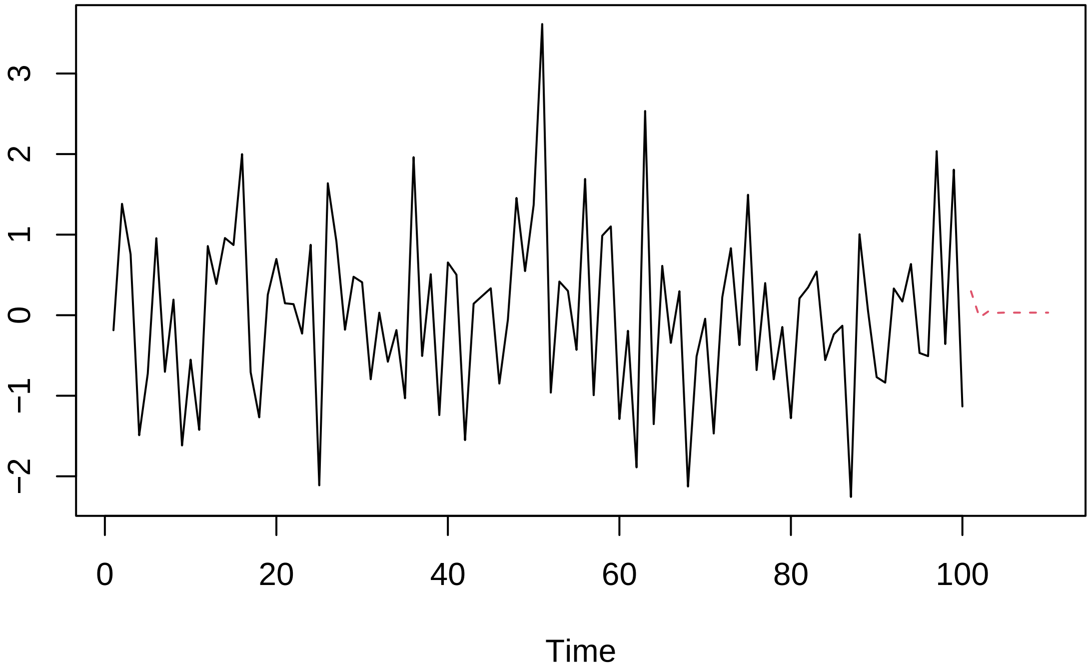
```
## [1] 1
```

```
## Parameter Confidence Interval
x.ar$ar + c(-1.96, 1.96) * sqrt(x.ar$asy.var.coef[1, 1])
```

```
## [1] -0.42041868 -0.03483392
```

```
## Predict 10 Steps Ahead
set.seed(832)
```

```r
x.ts <- ts(x); x.predict <- predict(x.ar, n.ahead = 1 * 10)

## Plot Series with Predictions
ts.plot(x.ts, x.predict$pred, col = 1:2, lty = 1:2)
```



The predictions leave much to be desired than those from part (c). The predictions with $\alpha = -0.4$ appear to stay constant, while those from part (c) appear to flucuate like the actual AR(1) process. This feels as if the function is scared to make a meaningful prediction on one side of the mean or the other.