STA 368 HW1 R Portion

Jack Liddicoat

2024-02-01

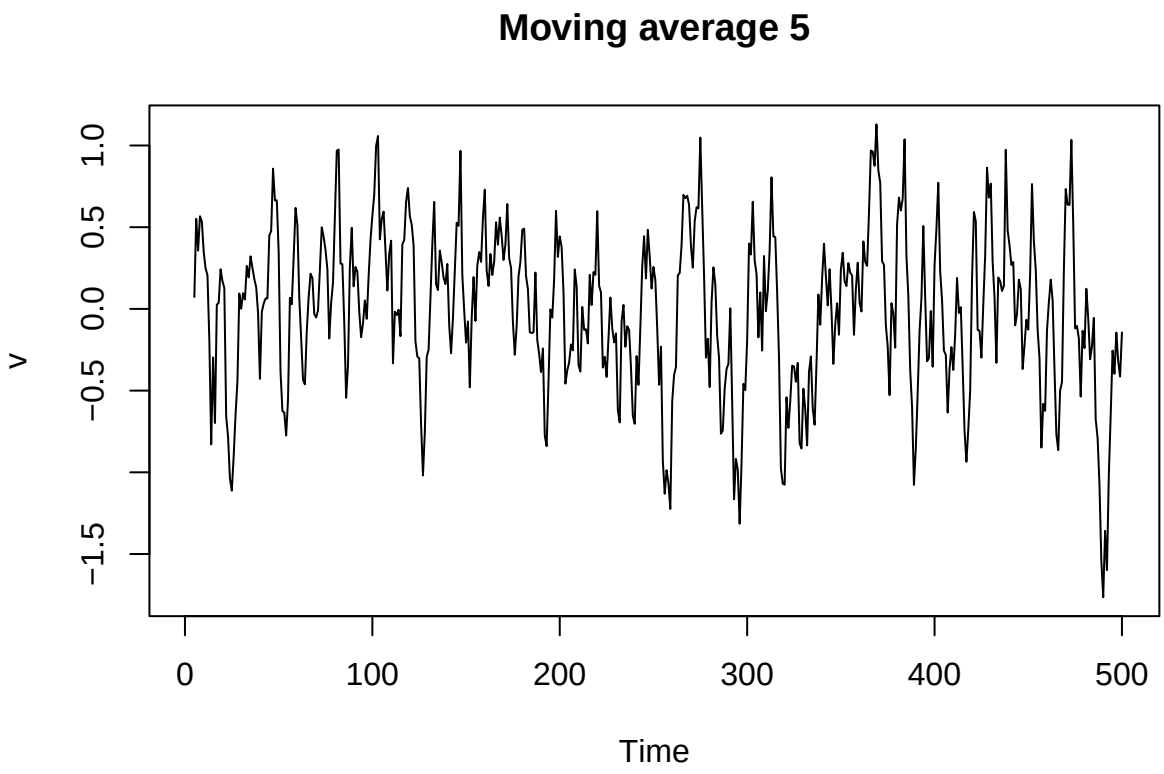
Sample Question

```
A = rnorm(12, 0, 1)
A
```

```
##  [1] -0.30053861 -0.70493509 -0.83780804 -2.39311152 -2.46873942  0.40417124
##  [7] -1.51675875 -1.28652272  2.11281801  0.73949400 -0.58642491  0.02669351
```

Problem 1

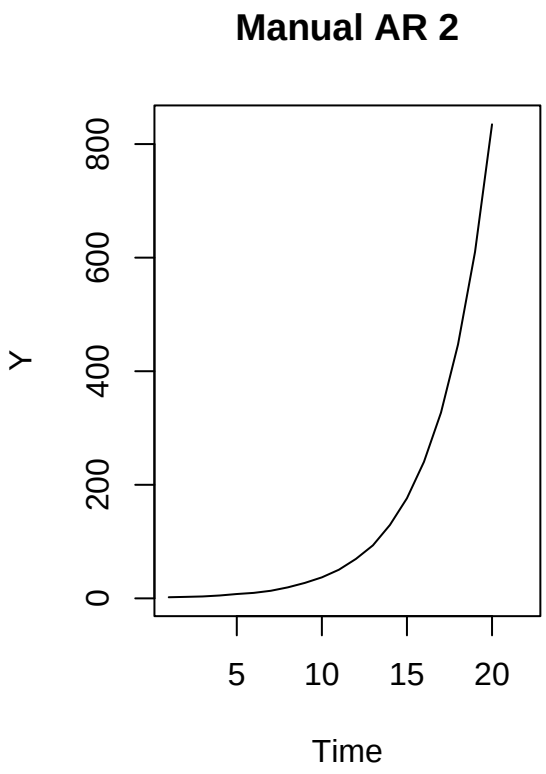
```
set.seed(1122)
my_maq <- function(n, q) {
  w <- rnorm(n, 0, 1)
  v <- filter(w, sides = 1, filter = rep(1/q, q))
  plot.ts(v, main = paste("Moving average", q))
}
my_maq(500, 5)
```

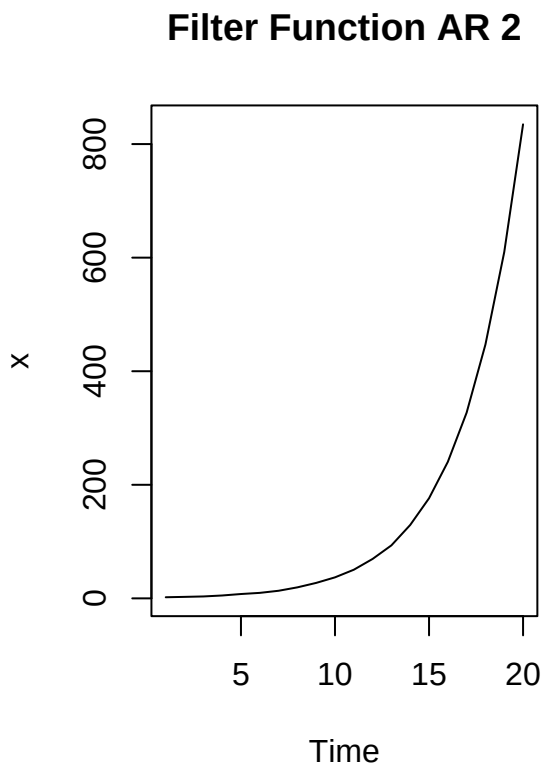


Problem 2

(a) and (b)

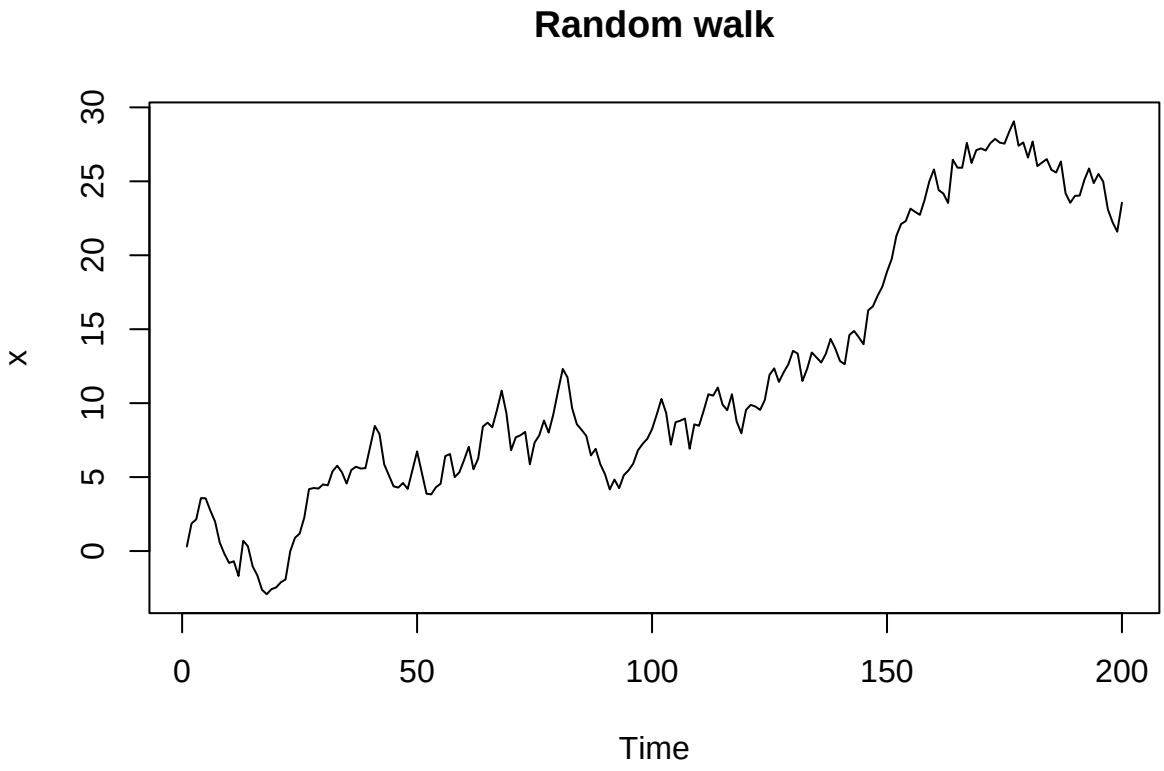
```
my_ar <- function(n, p) {
  Y <- vector(length = n)
  w <- rnorm(n, 0, 1)
  a <- vector(length = p)
  for (i in 1:p) {
    a[i] = 1/i
  }
  Y[1] <- w[1]
  for (i in 2:p) {
    Y[i] <- crossprod(a[1:(i-1)], Y[(i-1):1]) + w[i]
  }
  for (i in p+1:n) {
    Y[i] <- crossprod(a, Y[(i-1):(i-p)]) + w[i]
  }
  par(mfrow=c(1, 2))
  plot.ts(Y, main = paste("Manual AR", p))
  x = filter(w, filter = a, sides = 1, method = "recursive")
  plot.ts(x, main = paste("Filter Function AR", p))
}
my_ar(20, 2)
```





Problem 3

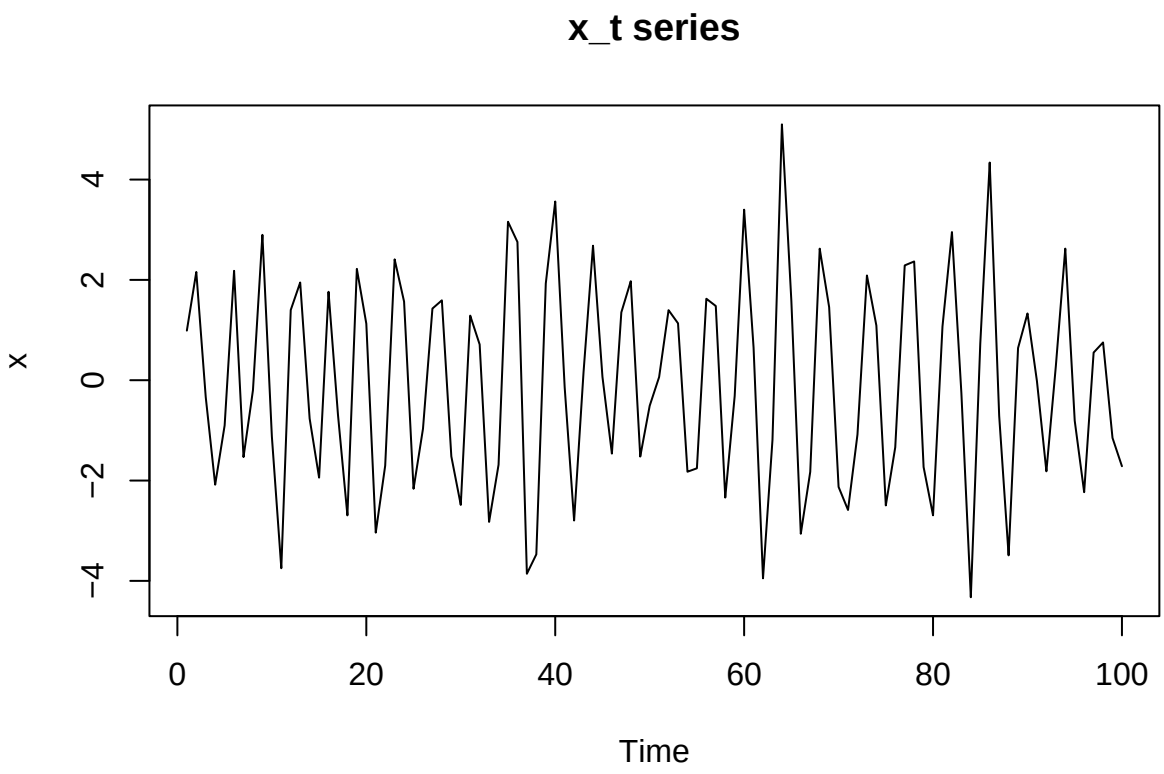
```
w = rnorm(200, 0, 1); x = cumsum(w)
wd = w + 0.1; x = cumsum(wd)
plot.ts(x, main = "Random walk")
```



Problem 4

(a)

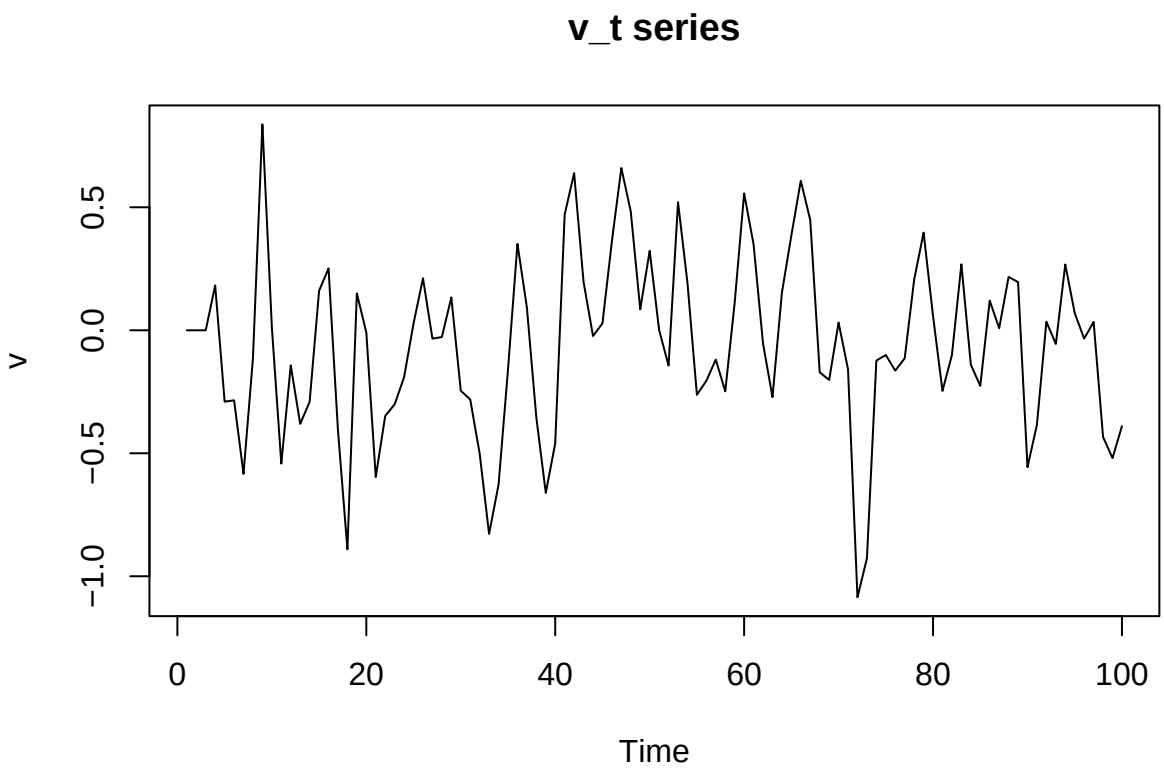
```
w = rnorm(100)
x = vector(length = 100)
for (t in 1:100) {
  if (t - 2 <= 0) {
    x[t] = w[t]
  }
  else {
    x[t] = -0.9*x[t-2] + w[t]
  }
}
plot.ts(x, main = "x_t series")
```



(b)

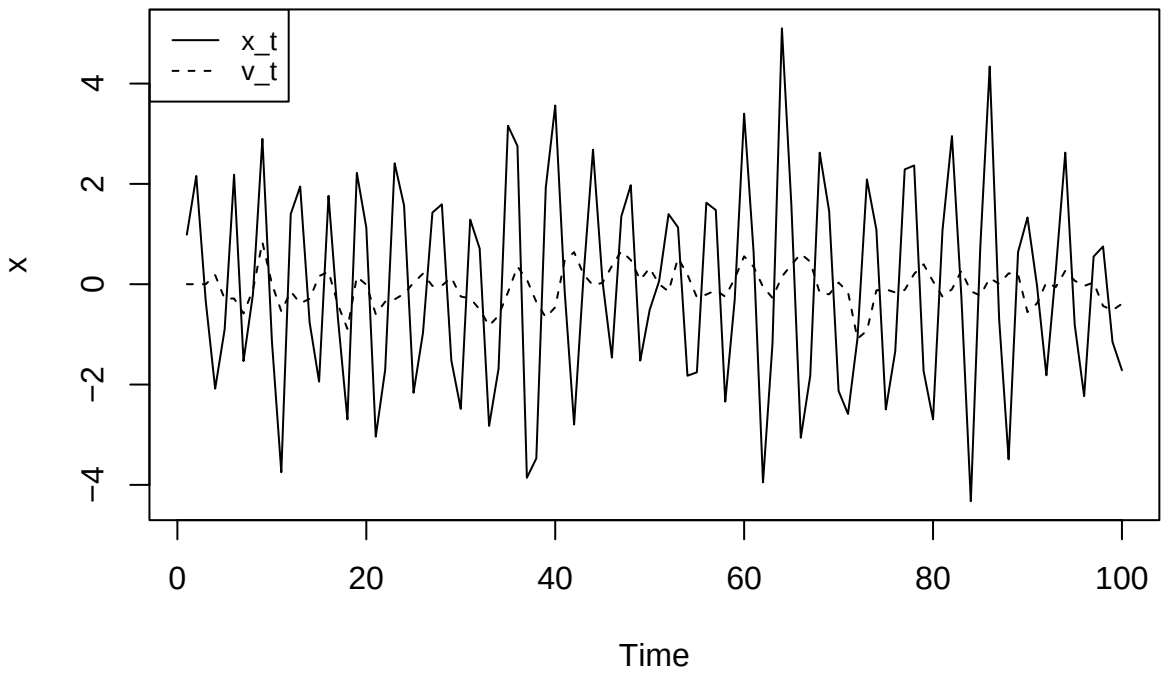
```
v = vector(length = 100)
for (t in 4:100) {
  v[t] = (x[t] + x[t-1] + x[t-2] + x[t-3])/4
}

plot.ts(v, main = "v_t series")
```



(c)

```
plot.ts(x)
lines(v, lty = "dashed")
legend("topleft", legend = c("x_t", "v_t"), lty = c(1:2), cex = .8)
```

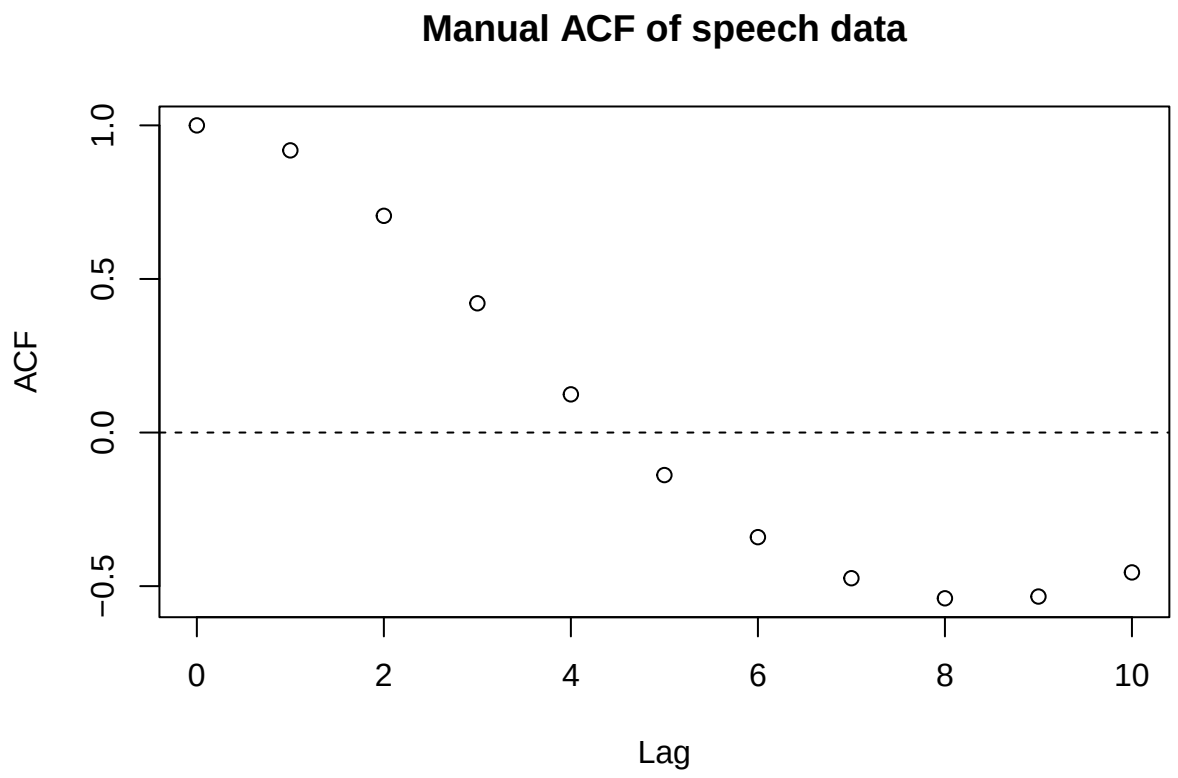


Problem 7

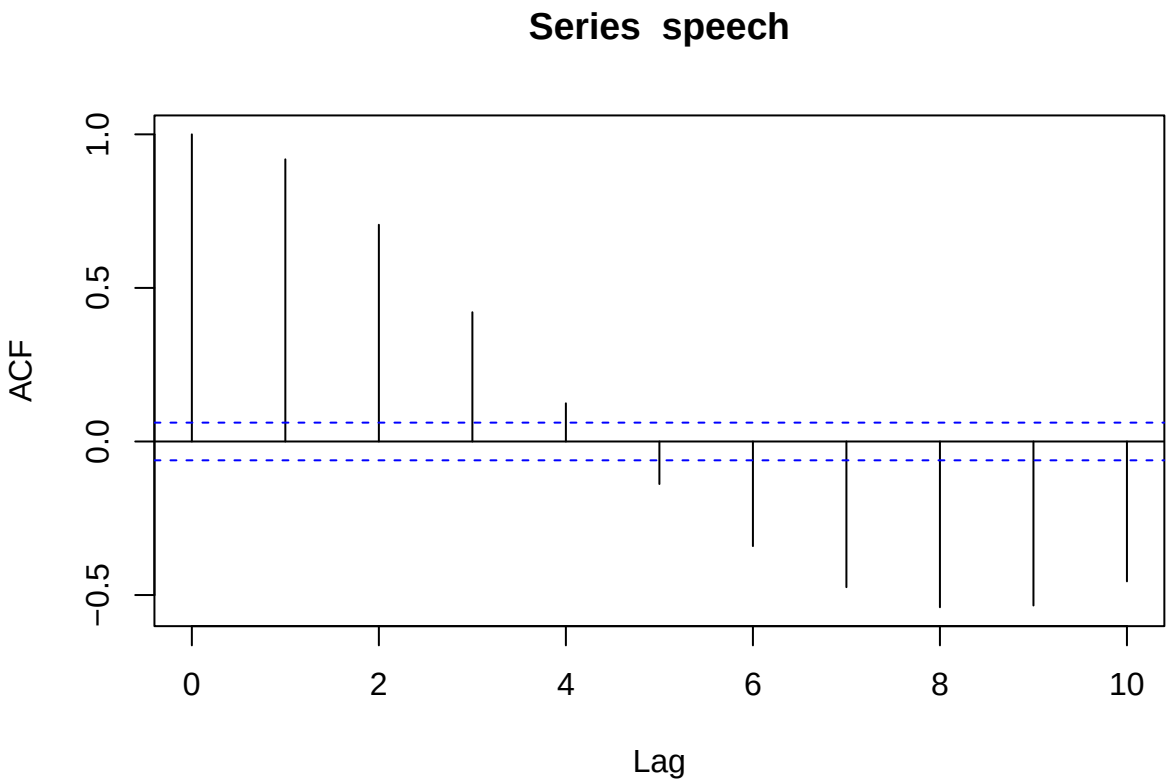
```
library(astsa)
acf_lag <- function(X, h) {
  coef <- vector(length = (h + 1))
  for (i in 1:(h+1)) {
    t <- 1
    n <- length(X)
    cov_sum <- 0
    while ((t + i) <= n) {
      s <- (X[t+(i-1)] - mean(X))*(X[t] - mean(X))
      cov_sum <- cov_sum + s
      t = t + 1
    }
    num <- cov_sum / n
    t <- 1
    s <- 0
    var_sum <- 0
    while (t <= n) {
      s <- (X[t] - mean(X))*(X[t] - mean(X))
      var_sum <- var_sum + s
      t = t + 1
    }
    denom <- var_sum / n
    auto_cor <- num / denom
    coef[i] <- auto_cor
```

```
  }
  lag <- seq(0, 10, by = 1)
  plot(coef ~ lag, main = "Manual ACF of speech data", xlab = "Lag", ylab = "ACF")
  abline(h = 0, lty = 2)
}

acf_lag(speech, h = 10)
```




```
acf(speech, lag.max = 10)
```



The plots are the same. In both of them the correlation is strongly positive at lags 0-2, but then they quickly turn negative at ~lag 5. After that, the ACF hits a nadir at lag 8 and moves towards 0 correlation. This looks very similar to a cosine function.