HW 6 - Jin Kweon (3032235207)

Jin Kweon

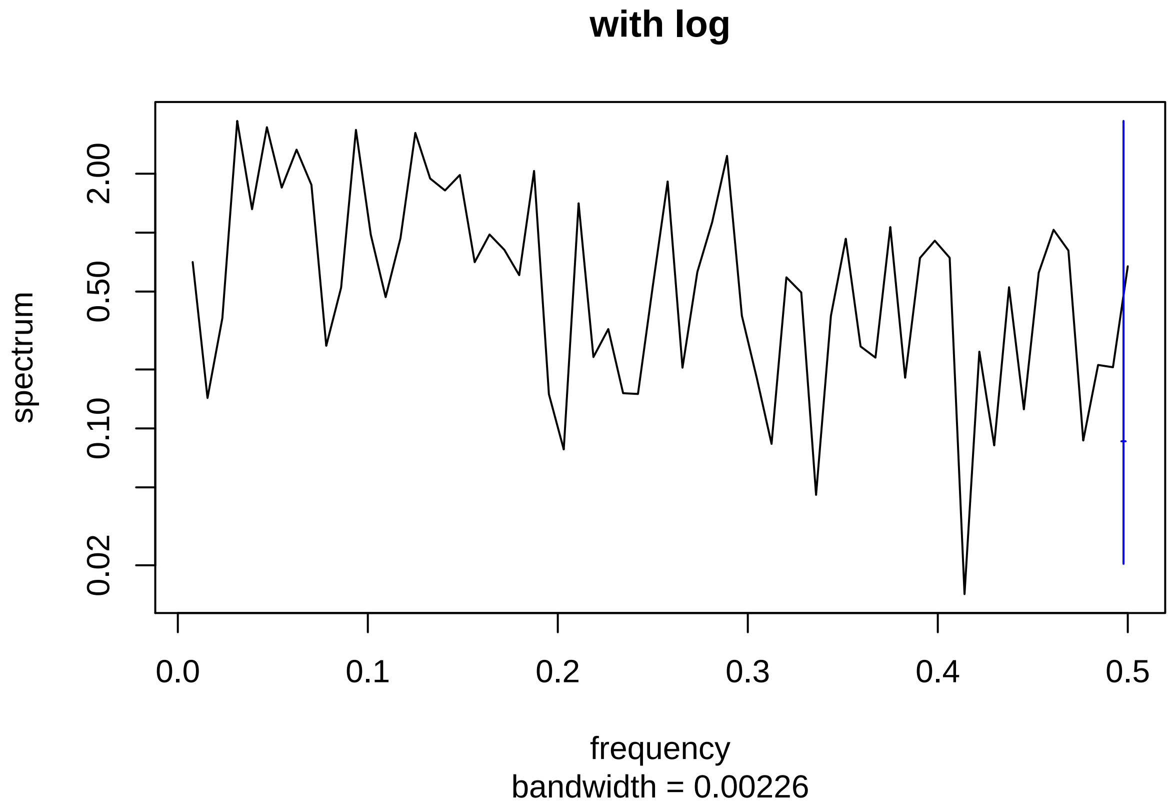
3/14/2018

1

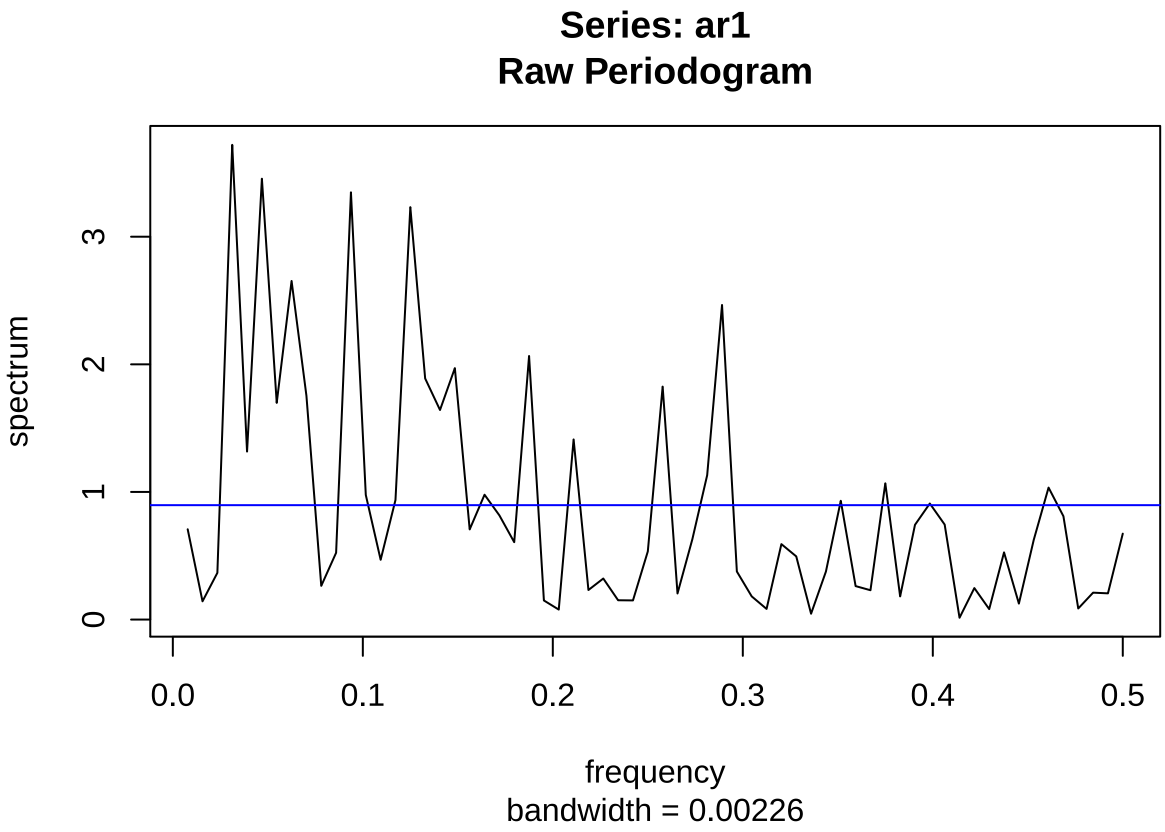
Good ref: https://onlinecourses.science.psu.edu/stat510/?q=book/export/html/57 & https://www.colorado.edu/geography/class_homepages/geog_4023_s09/TimeSeriesLab2.html

So, I basically need to do three things: generate, compute, and plot!

```
period <- function(data){
  n <- length(data)
  j <- 0:floor(n/2)
  freq <- j / n
  sumc <- 0
  sums <- 0

  for (i in j+1){
    sumc[i] <- sum(data * cos(2 * pi * freq[i] * (1:n)))
    sums[i] <- sum(data * sin(2 * pi * freq[i] * (1:n)))
  }

  output <- (sumc^2 + sums^2) / n

  return(list(output, freq))
}


set.seed(100)
# n = 128
n1 <- rnorm(128 + 50, 0, 1)
ar1 <- stats::filter(n1, filter = 0.5, method = "recursive")[-(1:50)]

#periodogram(ar1)
spec.pgram(ar1, main = "with log")
```

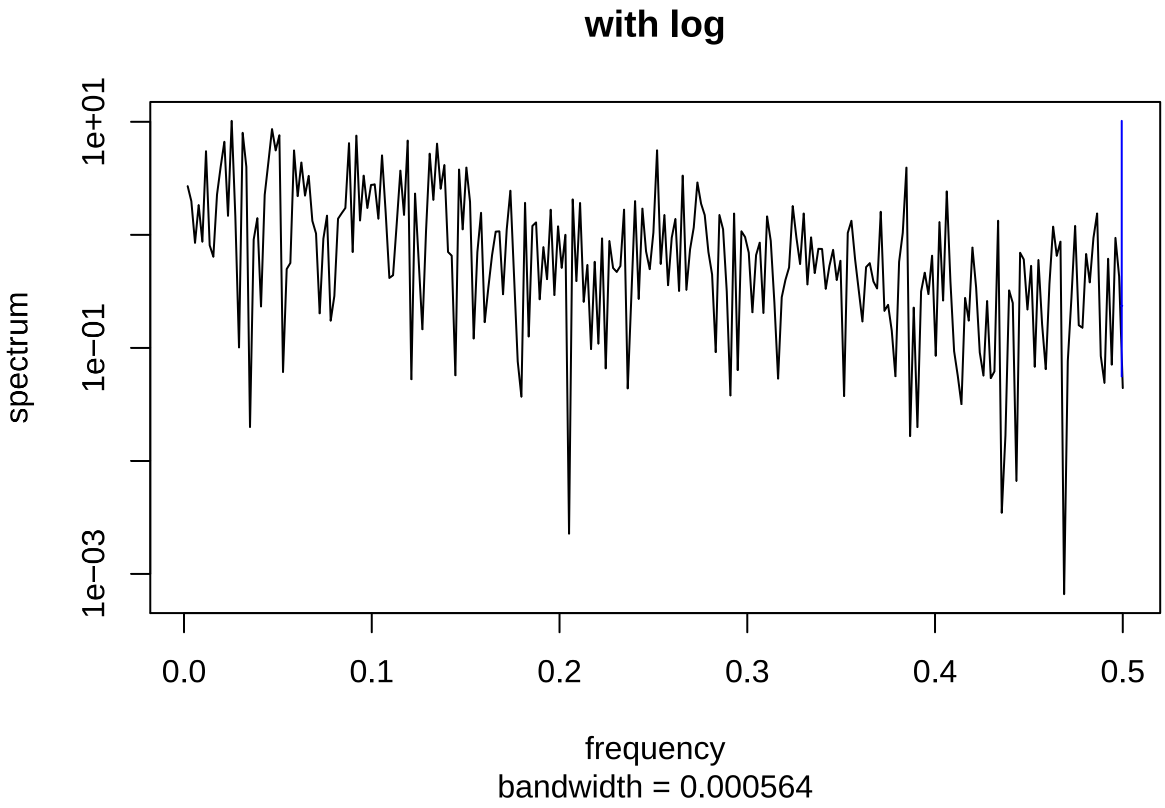


```r
spec1 <- spec.pgram(ar1, log = "no")
abline(h = mean(spec1$spec), col = "blue")
```

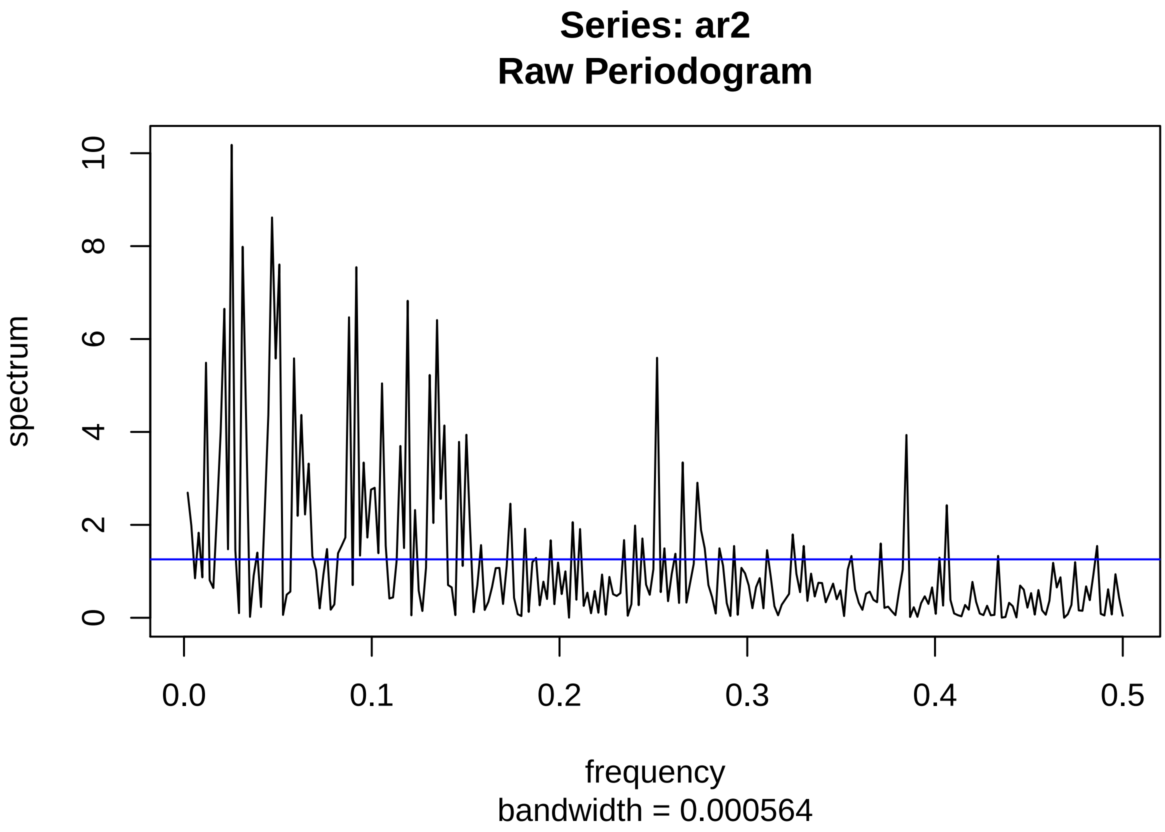


```r
data1 <- data.frame(freq = period(ar1)[[2]], spec = period(ar1)[[1]])
# ggplot(data1, aes(x = freq, y = spec)) + geom_path()
```

```r
set.seed(100)
# n = 512
n2 <- rnorm(512 + 50, 0, 1)
ar2 <- stats::filter(n2, filter = 0.5, method = "recursive")[-(1:50)]


#periodogram(ar2)
spec.pgram(ar2, main = "with log")
```

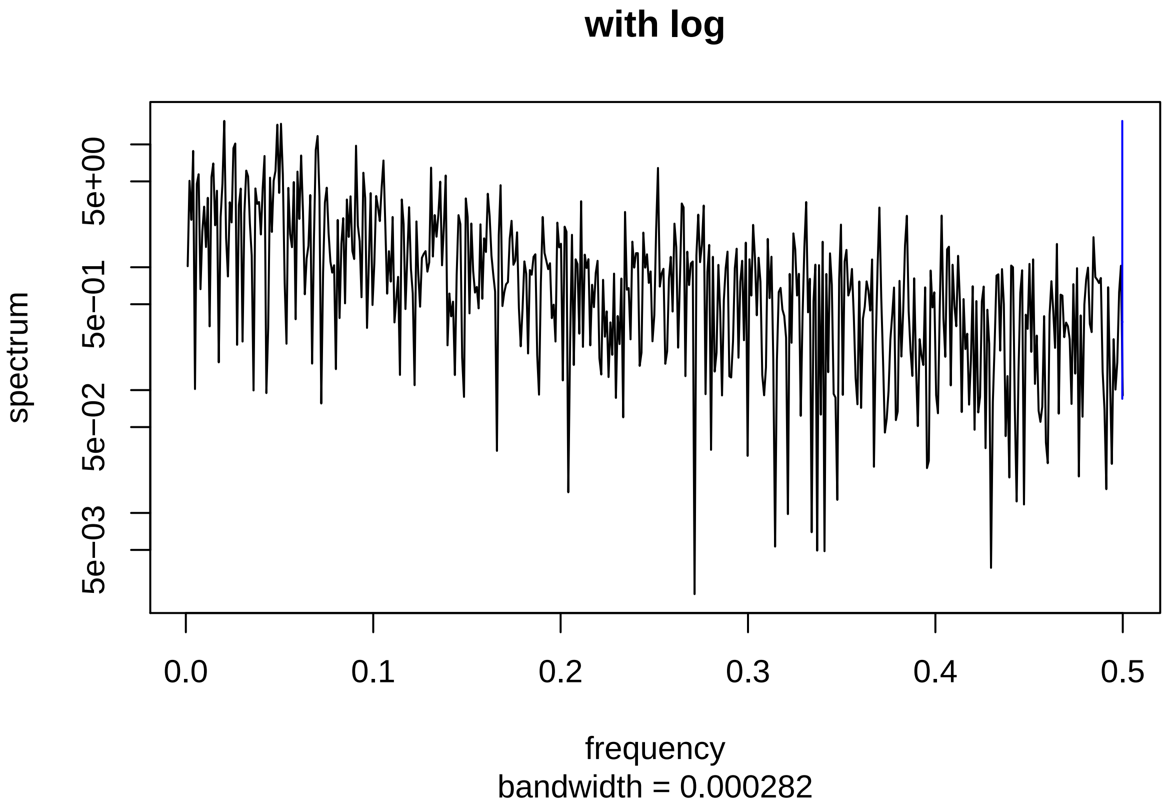


```
spec2 <- spec.pgram(ar2, log = "no")
abline(h = mean(spec2$spec), col = "blue")
```

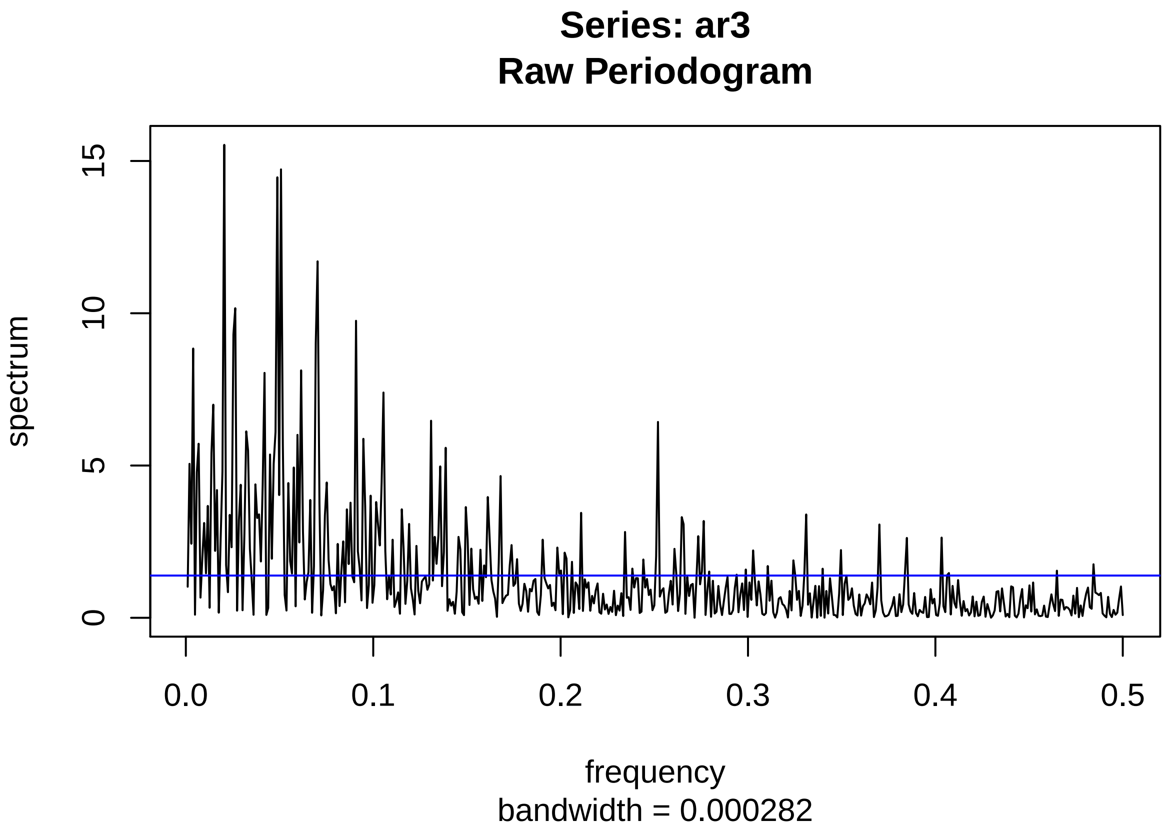


```
data2 <- data.frame(freq = period(ar2)[[2]], spec = period(ar2)[[1]])
# ggplot(data2, aes(x = freq, y = spec)) + geom_path()
```

```
set.seed(100)
# n = 1024
n3 <- rnorm(1024 + 50, 0, 1)
ar3 <- stats::filter(n3, filter = 0.5, method = "recursive")[-(1:50)]

#periodogram(ar3)
spec.pgram(ar3, main = "with log")
```

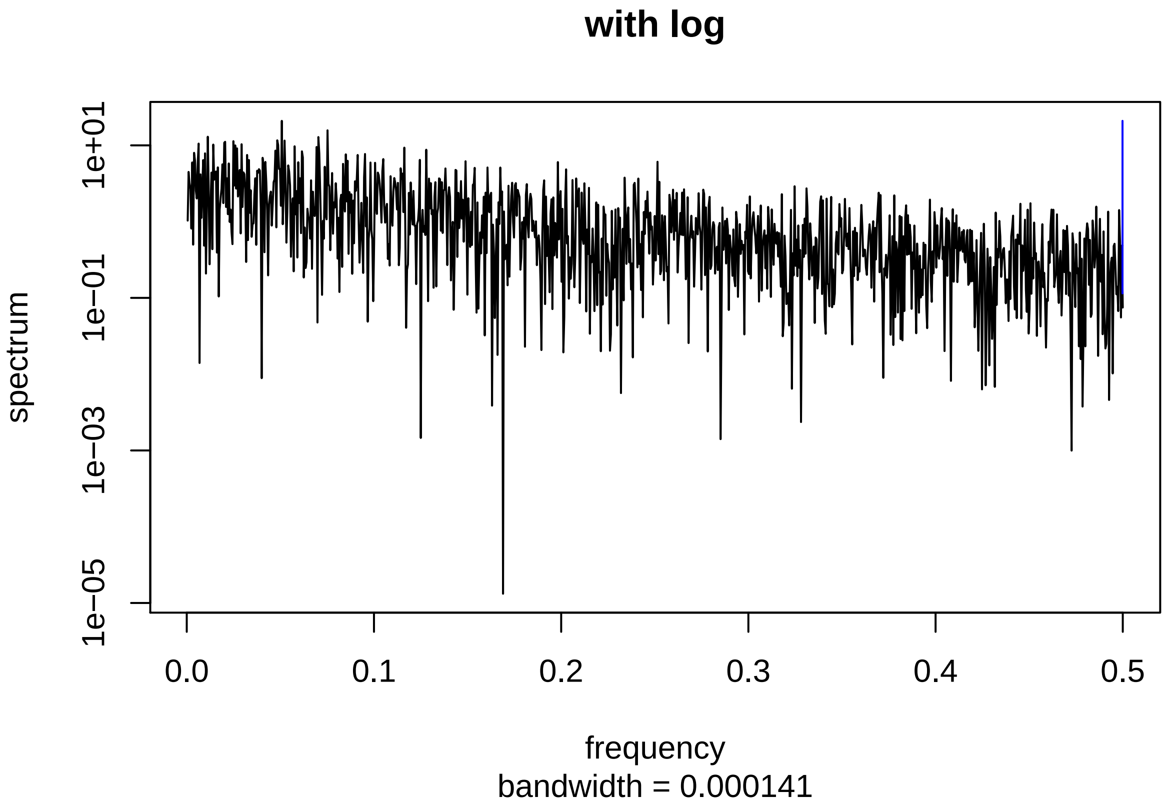


```r
spec3 <- spec.pgram(ar3, log = "no")
abline(h = mean(spec3$spec), col = "blue")
```

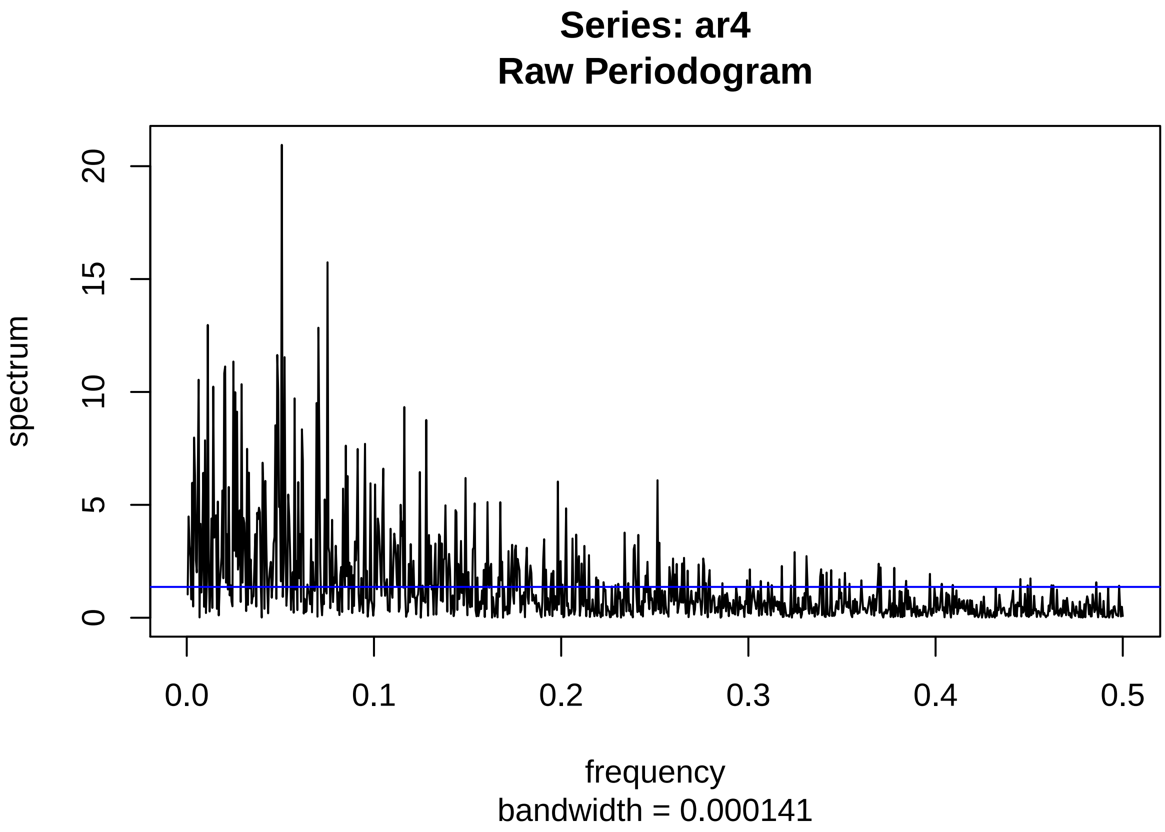


```r
data3 <- data.frame(freq = period(ar3)[[2]], spec = period(ar3)[[1]])
# ggplot(data3, aes(x = freq, y = spec)) + geom_path()
```

```r
set.seed(100)
# n = 2048
n4 <- rnorm(2048 + 50, 0, 1)
ar4 <- stats::filter(n4, filter = 0.5, method = "recursive")[-(1:50)]

#periodogram(ar4)
spec.pgram(ar4, main = "with log")
```



```r
spec4 <- spec.pgram(ar4, log = "no")
abline(h = mean(spec4$spec), col = "blue")
```



```
data4 <- data.frame(freq = period(ar4)[[2]], spec = period(ar4)[[1]])
# ggplot(data4, aes(x = freq, y = spec)) + geom_path()
```

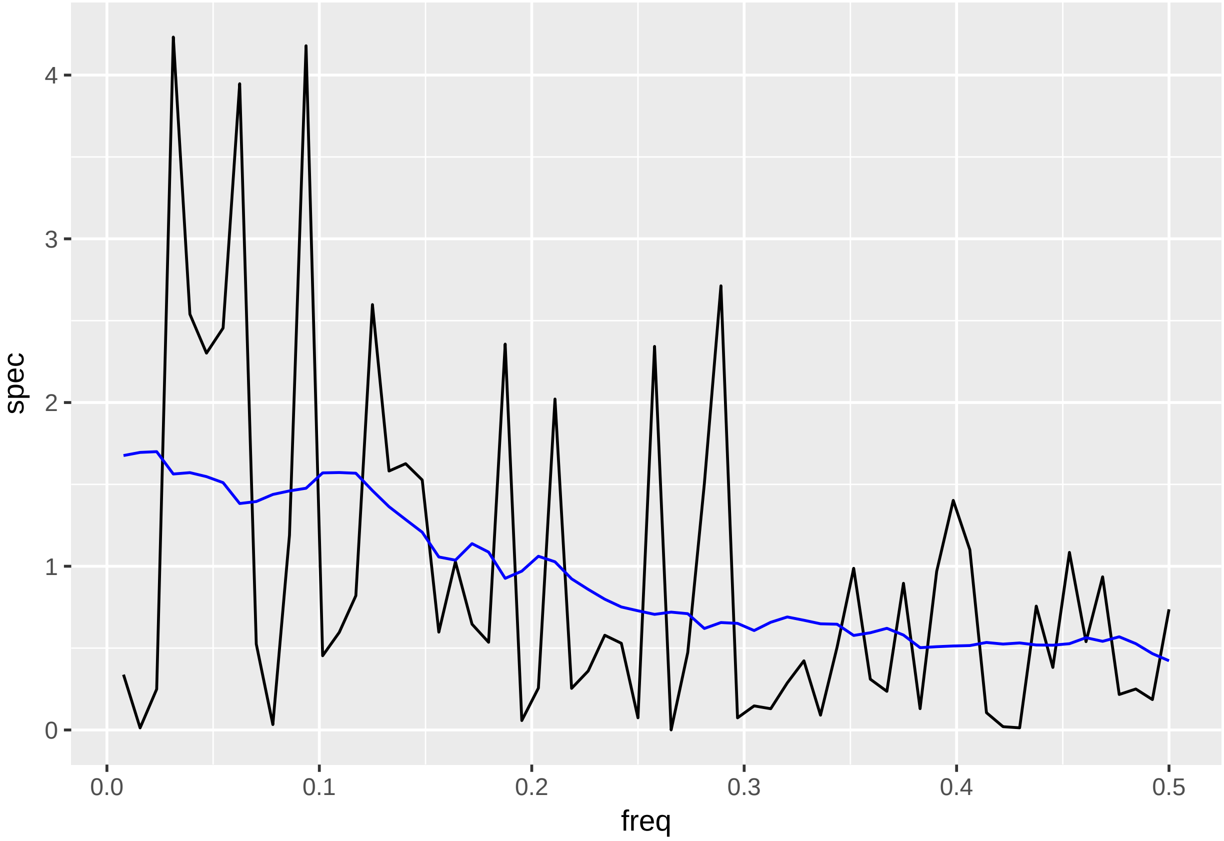
Comment:

I could catch a pattern better as I increase number of data.

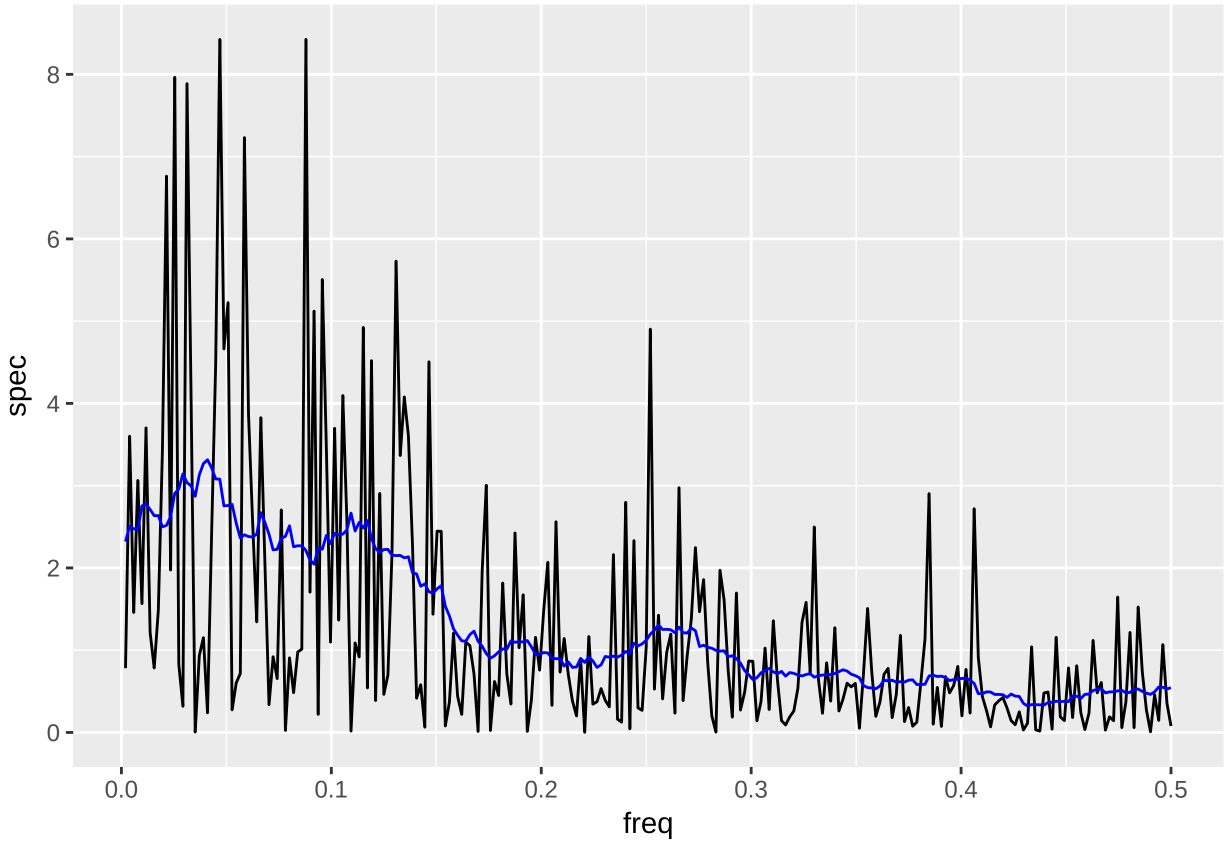
2

Use the Daniell Kernel

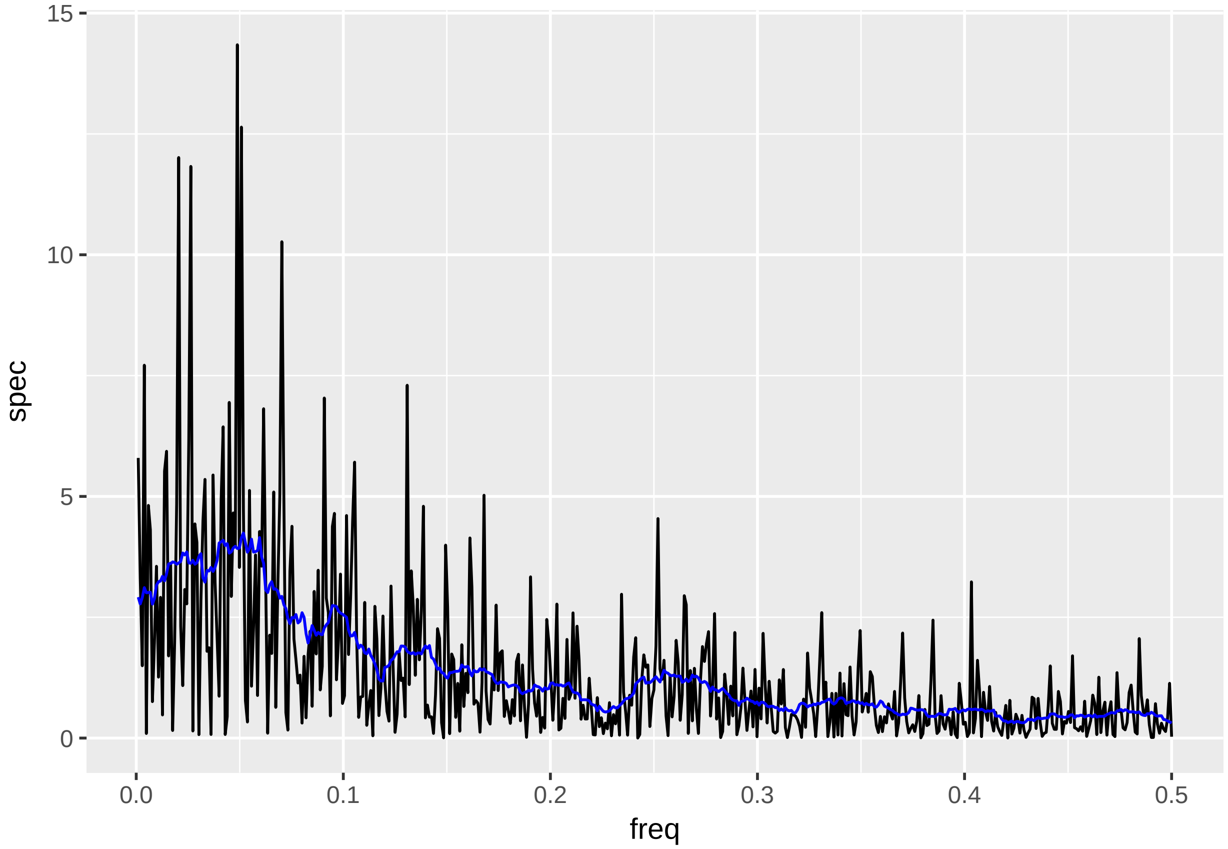
```
#n = 128
smooth1 <- cbind(data1[-1,],
                 smfreq = mvspec(ar1, kernel("daniell",floor(sqrt(length(ar1)))),
                                 log = "no", plot = F)$freq,
                 smspec = mvspec(ar1, kernel("daniell",floor(sqrt(length(ar1)))),
                                 log = "no", plot = F)$spec)

ggplot(smooth1) + geom_path(aes(x = freq, y = spec)) +
  geom_line(aes(x = smfreq, y = smspec), col = "blue")
```



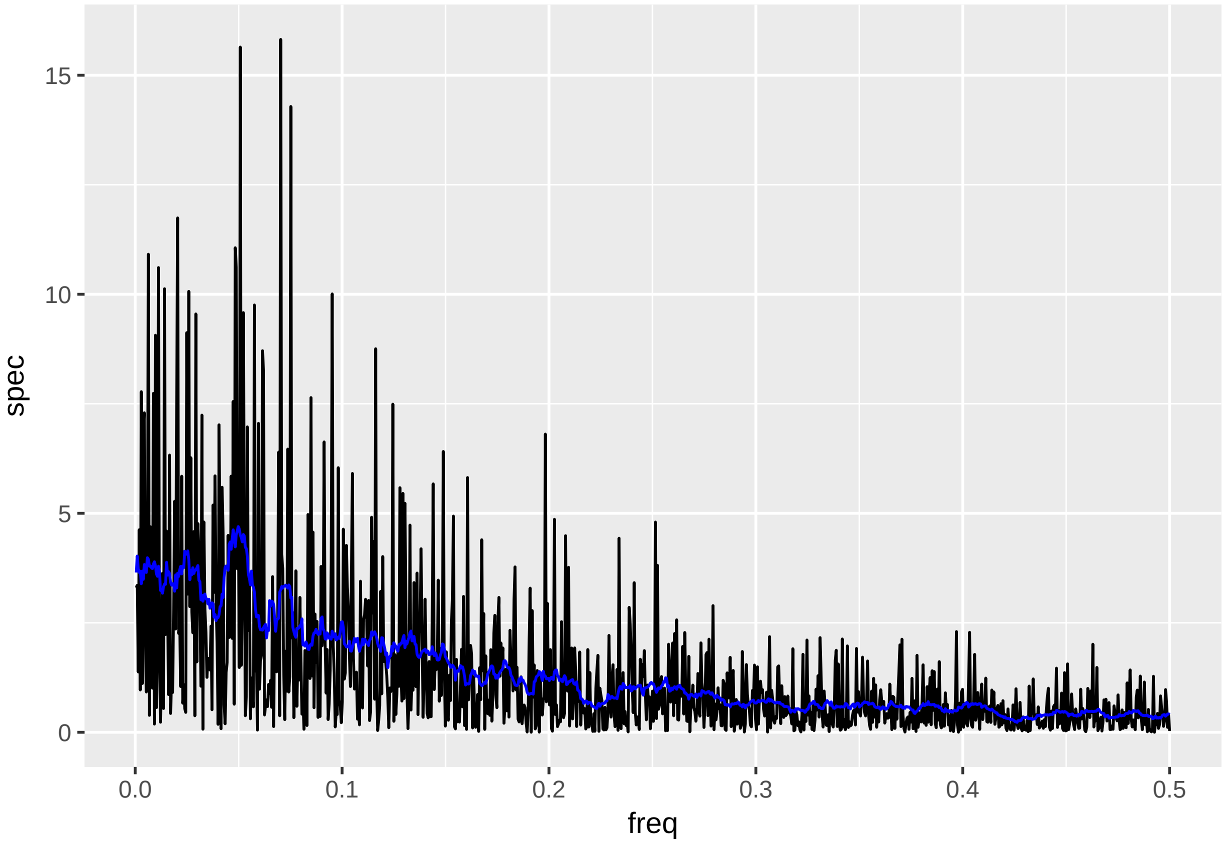
```
#or i can include kernel in spec.pgram



#n = 512
smooth2 <- cbind(data2[-1,],
                 smfreq = mvspec(ar2, kernel("daniell",floor(sqrt(length(ar1)))),
                                 log = "no", plot = F)$freq,
                 smspec = mvspec(ar2, kernel("daniell",floor(sqrt(length(ar1)))),
                                 log = "no", plot = F)$spec)

ggplot(smooth2) + geom_path(aes(x = freq, y = spec)) +
  geom_line(aes(x = smfreq, y = smspec), col = "blue")
```



```
#n = 1024
smooth3 <- cbind(data3[-1,],
                 smfreq = mvspec(ar3, kernel("daniell",floor(sqrt(length(ar1)))),
                                log = "no", plot = F)$freq,
                 smspec = mvspec(ar3, kernel("daniell",floor(sqrt(length(ar1)))),
                                log = "no", plot = F)$spec)

ggplot(smooth3) + geom_path(aes(x = freq, y = spec)) +
  geom_line(aes(x = smfreq, y = smspec), col = "blue")
```



```
#n = 2048
smooth4 <- cbind(data4[-1,],
                 smfreq = mvspec(ar4, kernel("daniell",floor(sqrt(length(ar1)))),
                                 log = "no", plot = F)$freq,
                 smspec = mvspec(ar4, kernel("daniell",floor(sqrt(length(ar1)))),
                                 log = "no", plot = F)$spec)

ggplot(smooth4) + geom_path(aes(x = freq, y = spec)) +
  geom_line(aes(x = smfreq, y = smspec), col = "blue")
```



Comment:

Obviously, smoothed periodogram average out my normal periodogram.