Characterising recruitment variability in MYDAS

C. Minto - for discussion

Load the RAM Legacy database[1] version 4.44

```
load("DBdata.RData")
```

Find out which sprat stocks are in RAM, for example

```
## get taxonomic serial number
tsn <- taxonomy$tsn[taxonomy$scientificname == "Sprattus sprattus"]
(tsn <- unique(tsn))

## [1] " 161789"

## link to stock table
idx <- which(stock$tsn == tsn)
stock_df <- stock[idx,]
stock_df[, c("stockid", "stocklong")]

##                    stockid                       stocklong
## 1119            SPRAT22-32 Sprat ICES Baltic Areas 22-32
## 1120             SPRATIIIa  Sprat Kattegat and Skagerrak
## 1121               SPRATNS                 Sprat North Sea
## 1122            SPRATVIIde                     Sprat VIIde
## 1123 SPRATVI-VIIabcfghijk     Sprat VI and VIIabcfghijk
## 1129           SPRBLKGSA29                 Sprat Black Sea
```

Get the stock-recruit data

```
stockids <- stock_df$stockid
## subset to these where recruitment data available
ts_values_df <- subset(timeseries_values_views, stockid %in% stockids & (!is.na(R)))
stockids <- unique(ts_values_df$stockid)

rownames(ts_values_df) <- NULL
head(ts_values_df[, c("stockid", "year", "SSB", "R")])

##      stockid year     SSB        R
## 1 SPRAT22-32 1973      NA 5.04e+10
## 2 SPRAT22-32 1974 1100000 1.89e+10
## 3 SPRAT22-32 1975  867000 1.94e+11
## 4 SPRAT22-32 1976  738000 4.27e+10
## 5 SPRAT22-32 1977 1260000 1.52e+10
## 6 SPRAT22-32 1978  866000 3.05e+10
```

[1] https://www.ramlegacy.org/database/

```
## these are lagged to the year spawned

## get the units
ts_units_df <- subset(timeseries_units_views, stockid %in% stockids)
rownames(ts_units_df) <- NULL
ts_units_df <- ts_units_df[, c("stockid", "SSB", "R")]

ts_units_df

##        stockid SSB   R
## 1   SPRAT22-32  MT E00
## 2      SPRATNS  MT E00
## 3 SPRBLKGSA29  MT E00

## ssb in metric tonnes and recruitment in numbers
```

Recruitment

Recruitment time series (Figure 1) display

- Similar scale despite no attempt to standardize for area - North Sea is approximately twice the area of either the Black Sea or Baltic - could look into this further to standardise for area and hence do something with more biological information.
- Relatively constrained, in the 10 or 100 billions, notwithstanding the drop for North Sea when also SSB dropped.
- Non-stationary at least in the mean for Baltic and North Sea sprat.
- Local 'spikes' with some evidence of pulses the periodicity of which can be debated.

```
library(ggplot2); theme_set(theme_bw())
ggplot(ts_values_df, aes(x = year, y = R)) +
    geom_line() +
    facet_wrap(~stockid, ncol = 1) +
    xlab("Year") +
    ylab("Recruitment")
```

Spectral analysis

Loosely a power spectrum tells us the contribution of each frequency to the overall process. This is useful in identifying periodicities and other structures in the time series. To analyse non-stationary random functions, we may proceed in at least two ways: transform the data to stationarity and process with classical spectral analysis or perform some form of time-dependent spectral analysis (see Priestley).
We first detrend the raw series so that they are approximately stationary in the first moment (Figure 2).

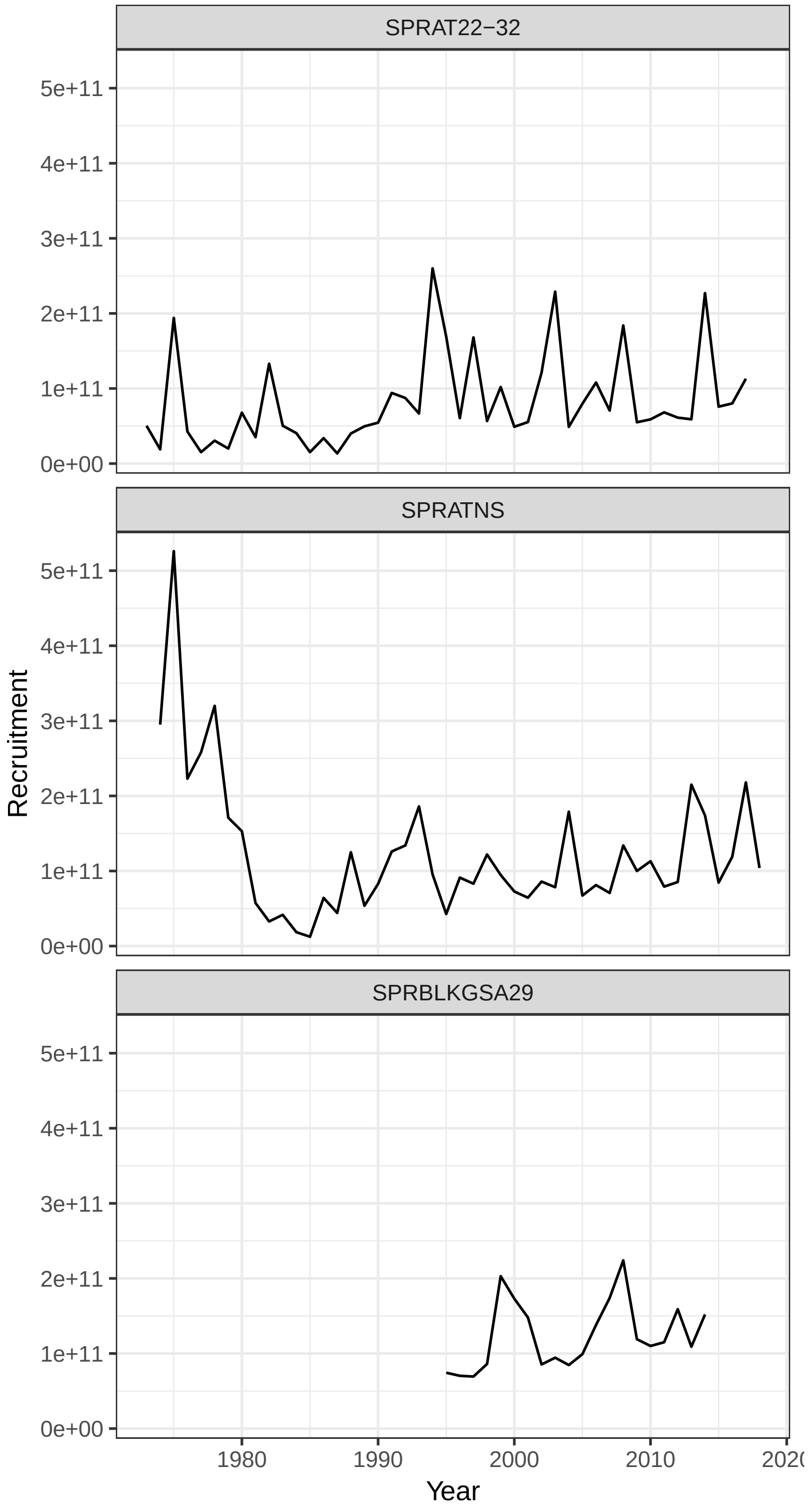

Figure 1: Sprat recruitment time series from the RAM Legacy database. Recruitment is in numbers.

Note that there are various theorems (Grenander and Rosenblatt) proved on the efficiency of spectral estimators with pre-treatment of the data with polynomials and then estimating the spectrum. This likely does not apply to very flexible local polynomials but here we will use relatively inflexible local polynomials that improve on global polynomials with inherent symmetry. We recognise though that by increasing the flexibility of the spline we may remove important periodicity in such short series.

```
library(mgcv)
library(gridExtra)
smoothed_df <- NULL

## fitting on natural log-scale
ts_values_df$logR <- log(ts_values_df$R)

for(i in 1:length(stockids)){
    dat <- subset(ts_values_df, stockid == stockids[i])
    ##fit <- gam(logR ~ s(year, k = 3), data = dat)
    fit <- gam(logR ~ s(year, k = 7), data = dat)
    tmp <- data.frame(stockid = stockids[i],
                      year = dat$year,
                      logR = dat$logR,
                      fit = predict(fit),
                      resid = resid(fit))
    smoothed_df <- rbind(smoothed_df, tmp)
}

raw_plots <- ggplot(smoothed_df, aes(x = year, y = logR)) +
    geom_line(colour = "slategrey") +
    geom_line(aes(y = fit)) +
    facet_wrap(~stockid, ncol = 1)

resid_plots <- ggplot(smoothed_df, aes(x = year, y = resid)) +
    geom_line(colour = "slategrey") +
    facet_wrap(~stockid, ncol = 1)

grid.arrange(raw_plots, resid_plots, ncol = 2)
```

Have a look at the stationarity of the residuals.

```
library(tseries)
for(i in 1:length(stockids)){
    print(stockids[i])
    print(adf.test(smoothed_df$resid[smoothed_df == stockids[i]]))
}

## [1] "SPRAT22-32"
##
##  Augmented Dickey-Fuller Test
##
## data:  smoothed_df$resid[smoothed_df == stockids[i]]
## Dickey-Fuller = -3.9863, Lag order = 3, p-value = 0.01889
## alternative hypothesis: stationary
##
## [1] "SPRATNS"
```

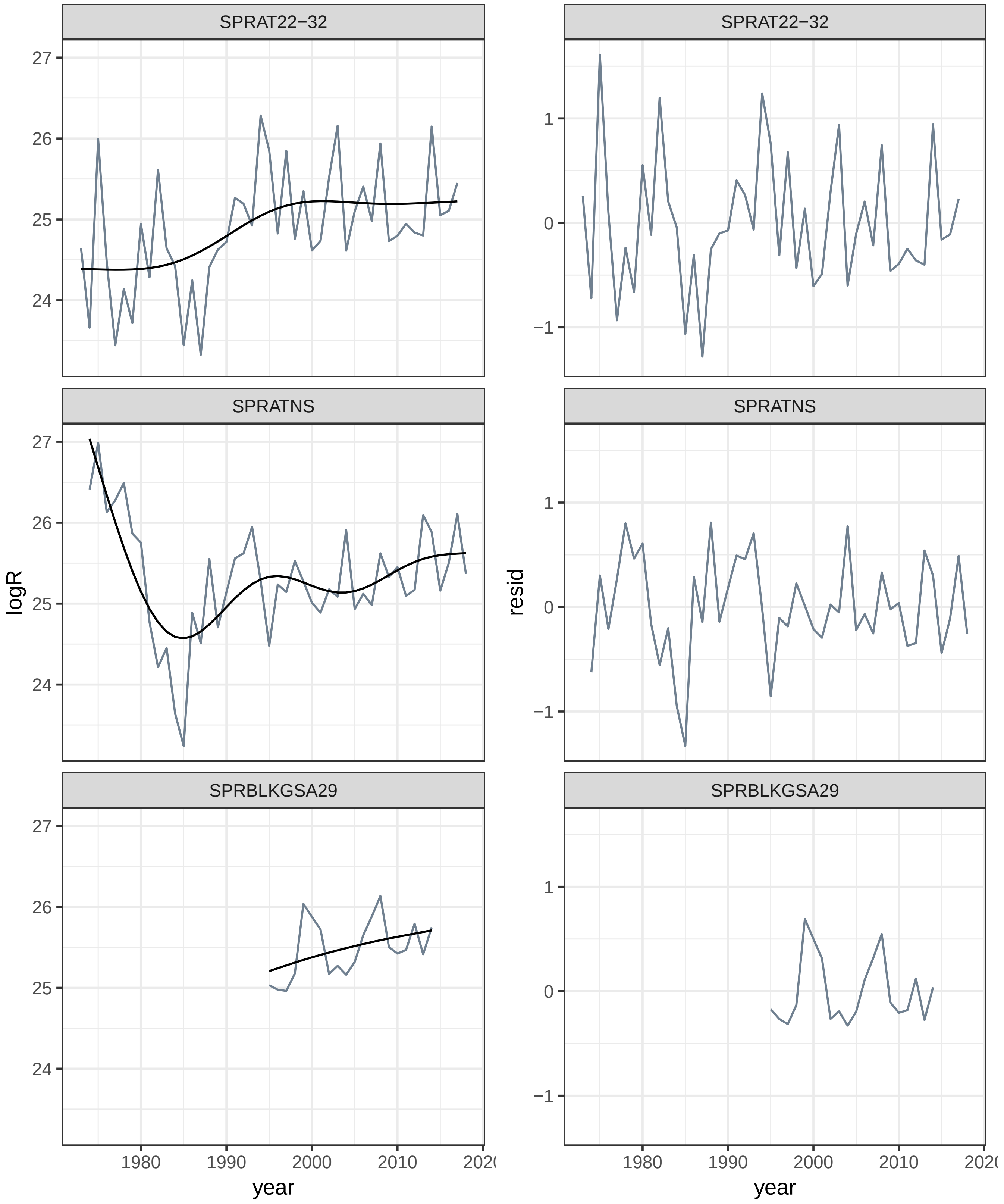

Figure 2: De-trending the natural logarithm of sprat recruitment using a low order basis smoothing spline.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  smoothed_df$resid[smoothed_df == stockids[i]]
## Dickey-Fuller = -3.6101, Lag order = 3, p-value = 0.0431
## alternative hypothesis: stationary
##
## [1] "SPRBLKGSA29"

## Warning in adf.test(smoothed_df$resid[smoothed_df == stockids[i]]):  p-value smaller
than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  smoothed_df$resid[smoothed_df == stockids[i]]
## Dickey-Fuller = -4.4663, Lag order = 2, p-value = 0.01
## alternative hypothesis: stationary
```

There are various algorithms to conduct an empirical spectral analysis on a time series. As the series are short, here we simply estimate a raw periodogram using FFT with no smoothing or tapering to see what periods dominate the series.

```
spec_df <- data.frame()

for(i in 1:length(stockids)){
    dat <- subset(smoothed_df, stockid == stockids[i])
    ## as the data so short no smoothing of spectrum
    fit  <- spec.pgram(dat$resid, taper = 0, log = "no", plot = FALSE)
    tmp <- data.frame(stockid = stockids[i],
                      freq = fit$freq,
                      period = 1/fit$freq,
                      spec = fit$spec
                      )
    spec_df <- rbind(spec_df, tmp)
}
```

Visualise the raw spectra (Figure 3)

```
ggplot(spec_df, aes(x = period, y = spec)) +
    geom_linerange(aes(ymin = 0, ymax = spec)) +
    facet_wrap(~stockid, ncol = 1) +
    ylab("Spectral density") +
    xlab("Period (years)")
```

High frequency (low period) variability dominates the Baltic and North Sea, superimposed on spectral peaks at approximately an 11 year period.

Next steps:

- Look at smoothing kernels and tapering

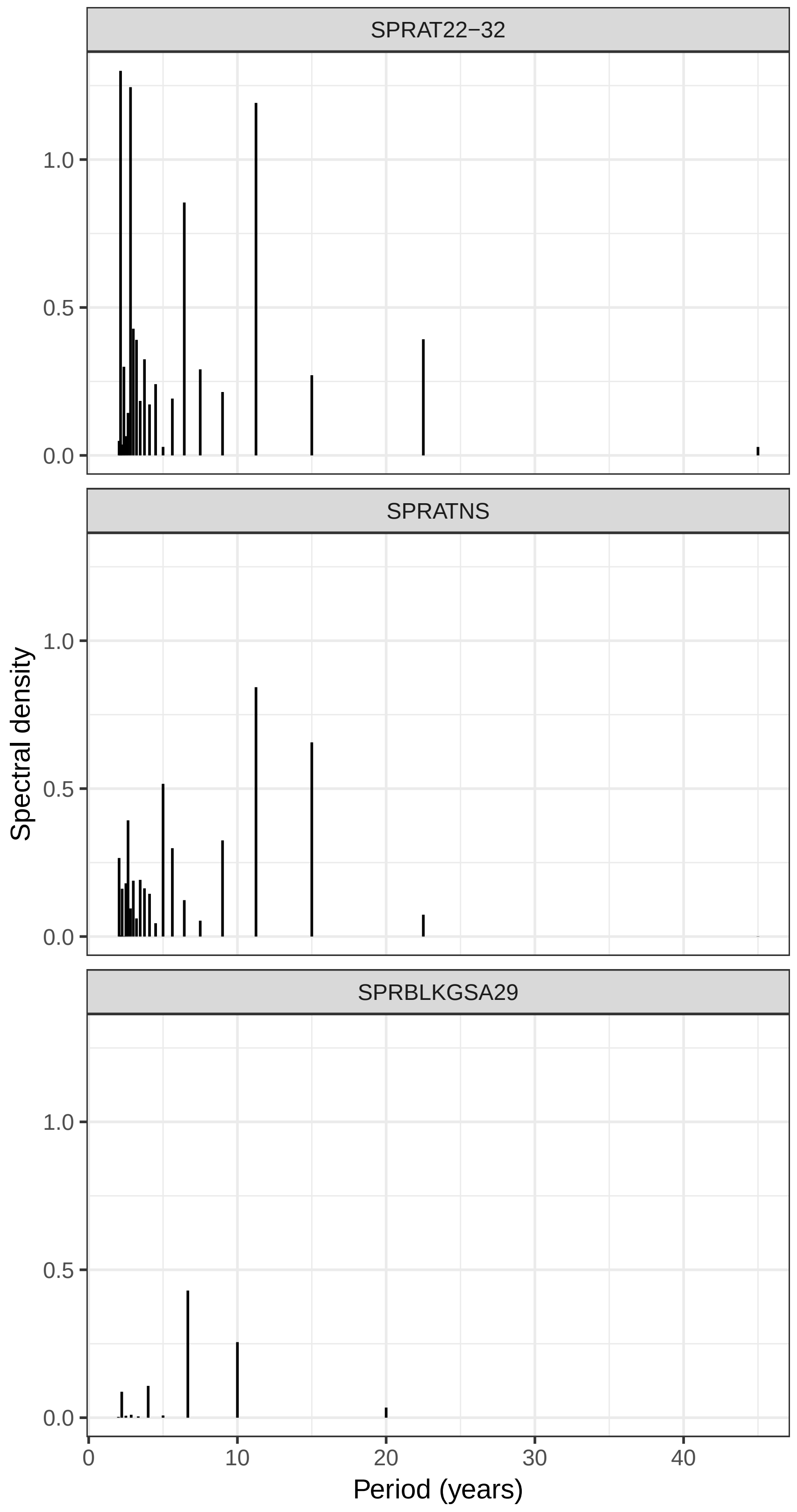

Figure 3: Raw periodograms for the de-trended sprat time series displaying the spectral density at given periods.

- Try a wavelet analysis for kicks, out of the blue and all that - too short here
- Simulate from a process closely matched to these spectra - see relationship of FFT with the coefficients of a harmonic regression.

Simulation

The periodogram value at frequency ω_j can be defined by the coefficients of a harmonic regression

$$I(\omega_j) = \frac{n}{4}(a_j^2 + b_j^2) \tag{1}$$

where a_j and b_j are the cosine and sine coefficients of the harmonic regression

$$y_t = \sum_{j=0}^{n/2} a_j \cos(\omega_j t) + b_j \sin(\omega_j t) \tag{2}$$

```
## fit the periodogram yourself
dat <- subset(smoothed_df, stockid == stockids[1])
fit  <- spec.pgram(dat$resid, taper = 0, log = "no", plot = FALSE, detrend = FALSE)

## set up the basis
n <- nrow(dat)
freq <- seq(1/n, by = 1/n, length.out = floor(n/2))
t <- 0:(n-1)
cos_x <- sapply(freq, function(f){cos(2 * pi * f * t)})
sin_x <- sapply(freq, function(f){sin(2 * pi * f * t)})

y <- dat$resid

fit_lm <- lm(y ~ -1 + cos_x + sin_x)

ab <- coef(fit_lm)

a <- ab[grep("cos_", names(ab))]
b <- ab[grep("sin", names(ab))]

## periodogram ordinate
Iom <- n/4 * (a^2 + b^2)

range(fit$spec - Iom)

## [1] -2.886580e-15  1.804112e-15
```

As the spectral representation is exact, if we predict forward in time we obtain a repeat of the time series (Figure 4)

```
## set up another "cycle"
plot(y, type = "l", xlim = c(0, 2 * n), bty = "l")
```

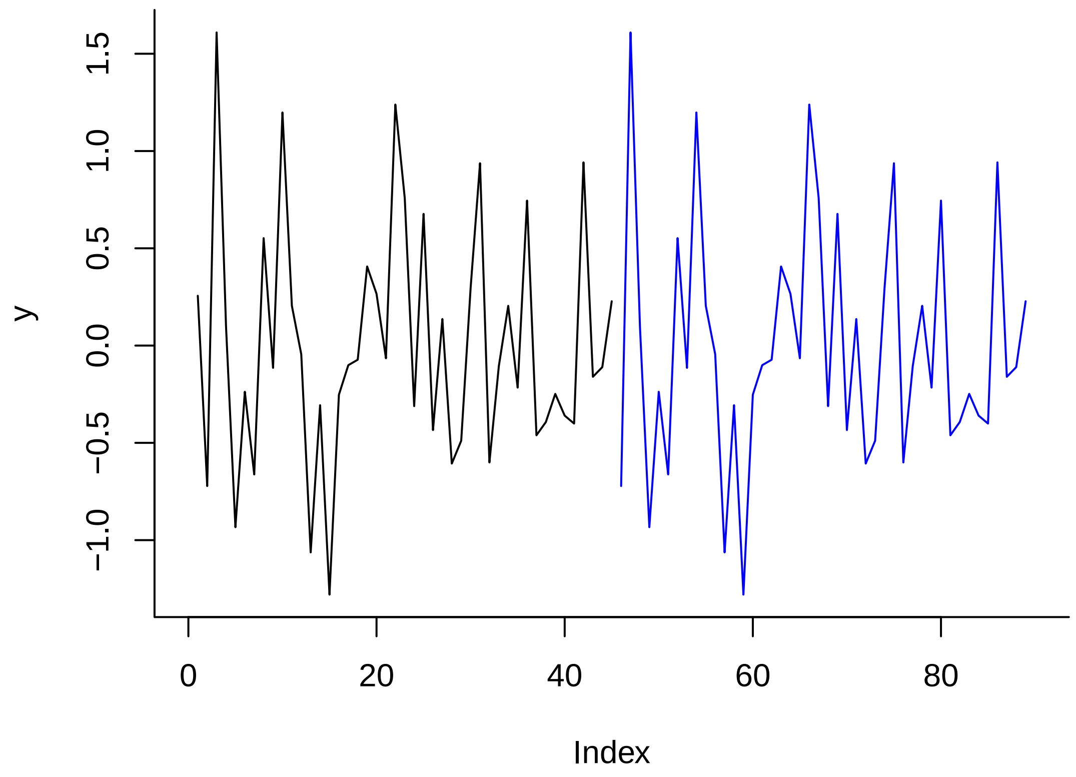

Figure 4: Prediction illustrating that the function repeats exactly under the raw periodogram.

```
t_new <- seq(n+1, 2 * n - 1)

cos_x <- sapply(freq, function(f){cos(2 * pi * f * t_new)})
sin_x <- sapply(freq, function(f){sin(2 * pi * f * t_new)})

y_pred <- c(cos_x %*% a + sin_x %*% b)

lines(t_new, y_pred, col = "blue")
```

We need an approach to introduce randomness. Here we introduce randomness by sampling the periodogram such that the probability a given frequency is chosen is proportional to its periodogram ordinate (Figure 5).

```
## think about this
p_Iom <- Iom / sum(Iom)

## control number of samples
m <- 50
period_sample <- 1 / sample(freq, size = m, prob = p_Iom, replace = TRUE)
```

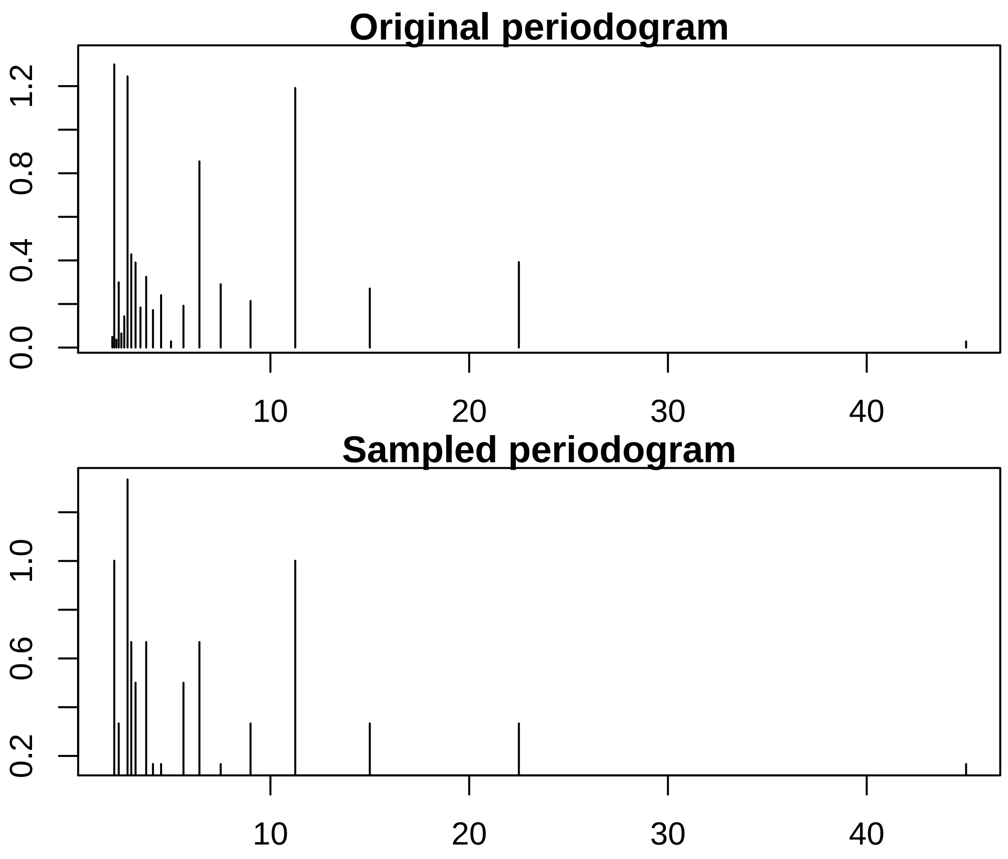

Figure 5: Preliminary sample of periodogram

```r
period_sample_tab  <- prop.table(table(period_sample)) * sum(Iom)

par(mfrow = c(2, 1), oma = c(2, 2, 1, 1), mar = c(2, 2, 1, 1))
plot(1/freq, Iom, type = "h", ylim = range(Iom, period_sample_tab),
     main = "Original periodogram")
plot(as.numeric(names(period_sample_tab)),
     as.numeric(period_sample_tab),
     type = "h", main = "Sampled periodogram", xlim = range(1/freq))
```