

# TimeSeriesData

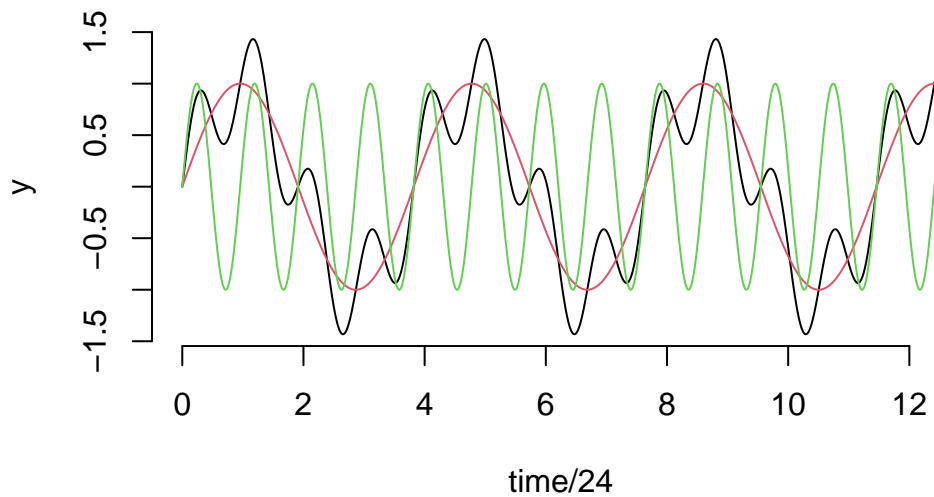
```
source("clarkFunctions2025.r")
```

```
time2radians <- function( time, period )time/period*2*pi

period1 <- 2*pi/24 # period 24 and 6 in radians
period2 <- 2*pi/6

time    <- seq( 0, 300, by = .1)
radians <- time2radians( time, 24 )

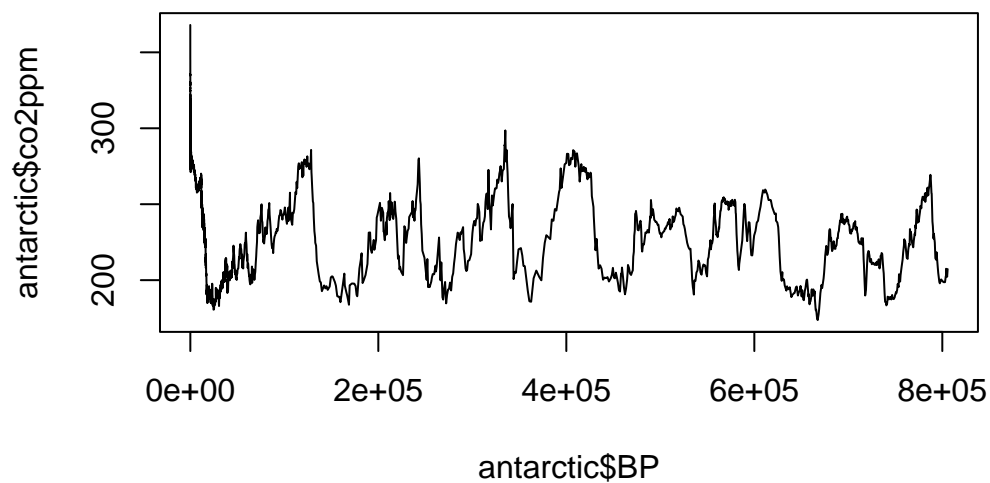
y1 <- sin( period1*radians )
y2 <- sin( period2*radians )
plot( time/24, 1*y1 + .5*y2, type = 'l', bty = 'n', ylab = 'y' )
lines( time/24, y1, col = 2 )
lines( time/24, y2, col = 3 )
```



```
antarctic <- read.table(
  'https://www1.ncdc.noaa.gov/pub/data/paleo/icecore/antarctica/antarctica2015co2composite.txt',
  skip = 138 )
```

1. Use the colnames function to give columns these names: 'BP', 'co2ppm', 'uncertainty'
2. Create a plot of co2ppm against years BP. Note that time runs from right to left. To turn it around, create a new column called year, as 1950 - BP.

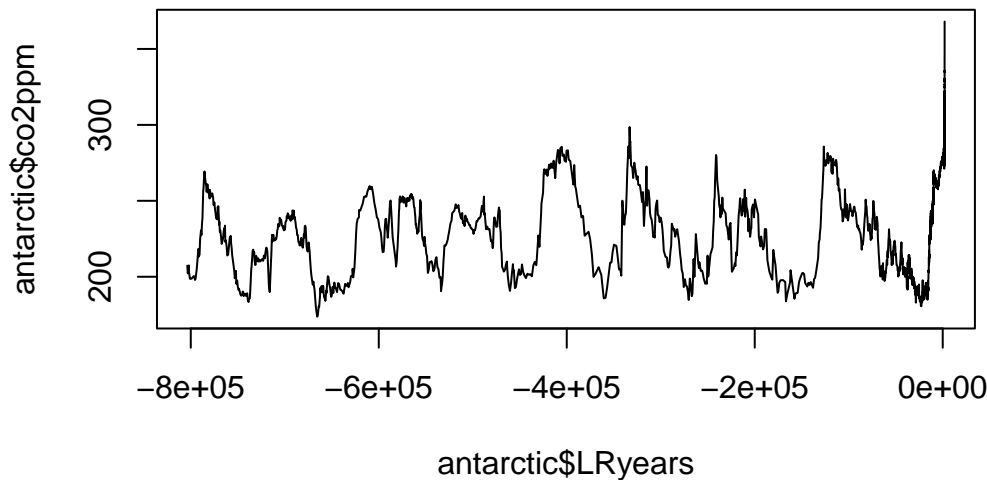
```
colnames(antarctic) <- c('BP', 'co2ppm', 'uncertainty')
plot(antarctic$BP, antarctic$co2ppm, type = 'l')
```



```
antarctic$LRyears <- 1950-antarctic$BP
```

3. Plot co2ppm against year.

```
plot(antarctic$LRyears, antarctic$co2ppm, type = 'l')
```



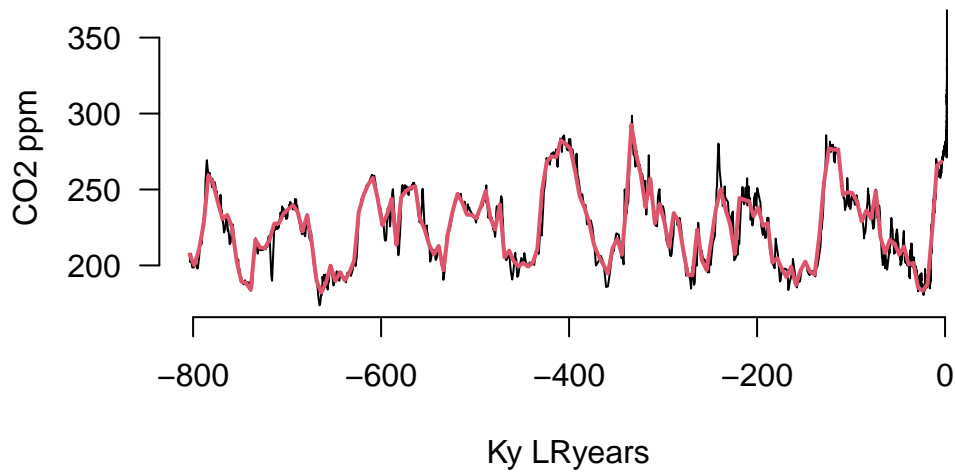
4. In groups, we will draft pseudocode to suggest how you might find the glacial maxima (low CO<sub>2</sub>) and minima (high CO<sub>2</sub>). For example, I would find the extreme values based on some criteria. Or something else. Together, we'll try to put this into R code.

I would probably wait until we do the weighted average data smoothing and then try to pull local min and max out. Right now, something like a differentiation approach has too many points.

5. In this example I do this: I create a sequence of years I find the closest interval for each year in the data (column BP) I determine how each point in the interval contributes to a value in years with a weighted average, where the weight that each point contributes to the evenly-spaced years is determined by their time differences I plot the original data with the new sequence:

```
years <- seq( min(antarctic$LRyears), max(antarctic$LRyears), by = 5000 )
iy <- findInterval( antarctic$LRyears, years, all.inside = T )
wt <- 1/(antarctic$LRyears - years[iy] + .001)^2 # time difference from closest yr
numerator <- tapply( antarctic$co2ppm*wt, iy, sum )
denominator <- tapply( wt, iy, sum )
cs <- numerator/denominator
years <- years[-1] - diff(years)[1]

plot( antarctic$LRyears/1000, antarctic$co2ppm, type = 'l', xlab = 'Ky LRyears',
      ylab = 'CO2 ppm', bty = 'n', las = 1 )
lines( years/1000, cs, lwd = 2, col = 2 )
```



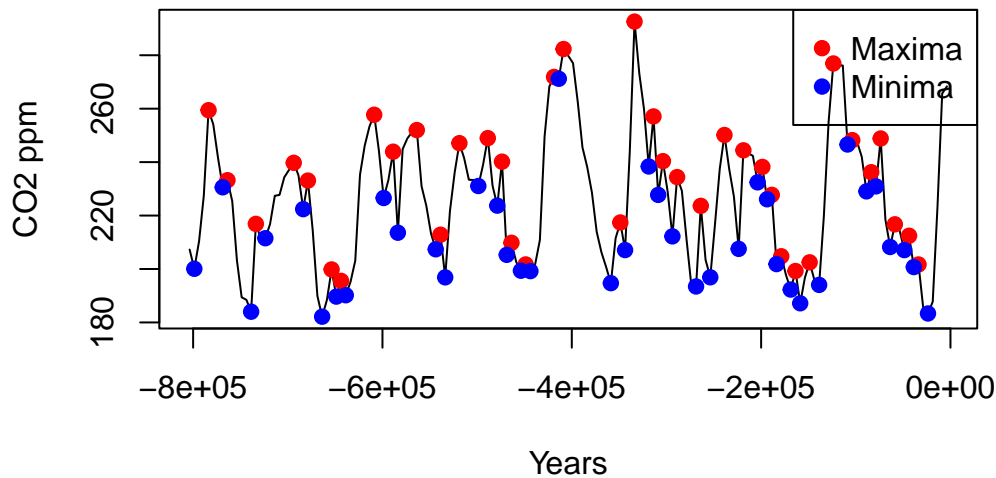
Going back to number 4 here, the smoothed data should lend itself to the original approach better with fewer data points, here is a look at all of the local min and max values, a few of these can be viewed as glacial min and max points, certainly not all of them.

```
# Calculate the difference
dx <- diff(cs)

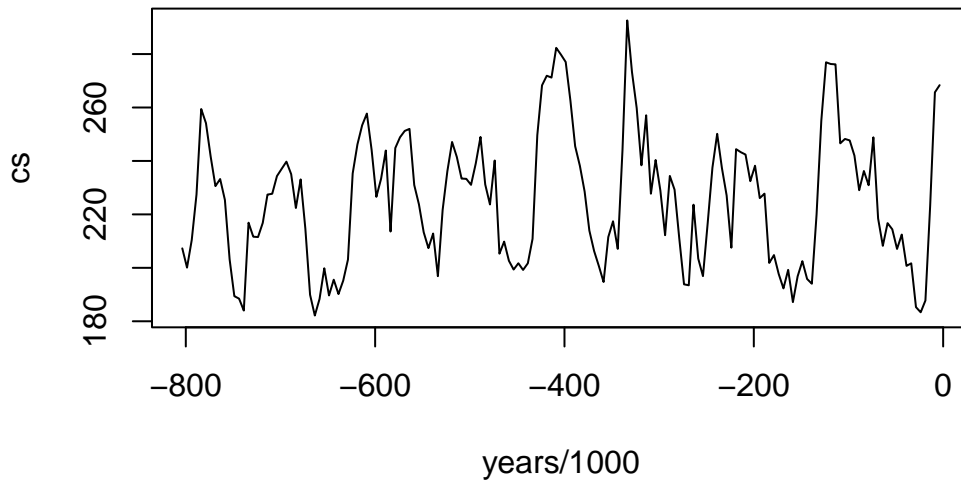
# Identify turning points
maxima_indices <- which(dx[-1] < 0 & dx[-length(dx)] > 0) + 1
minima_indices <- which(dx[-1] > 0 & dx[-length(dx)] < 0) + 1

# Extract corresponding years and values
maxima_years <- years[maxima_indices]
maxima_values <- cs[maxima_indices]
minima_years <- years[minima_indices]
minima_values <- cs[minima_indices]

# Plot
plot(years, cs, type = 'l', xlab = 'Years', ylab = 'CO2 ppm')
points(maxima_years, maxima_values, col = 'red', pch = 19)
points(minima_years, minima_values, col = 'blue', pch = 19)
legend("topright", legend = c("Maxima", "Minima"), col = c("red", "blue"), pch = 19)
```



```
plot(years/1000, cs, type = 'l')
```



6. How many “interglacials” are there over the last 800 Ky BP?

Looks like 9 interglacials, could be 7 or 8 depending on your interpretation of the graph, as 600 kya to 450 kya could be seen as one or two etc.

7. How do glacial periods compare with interglacials in terms of length and duration?

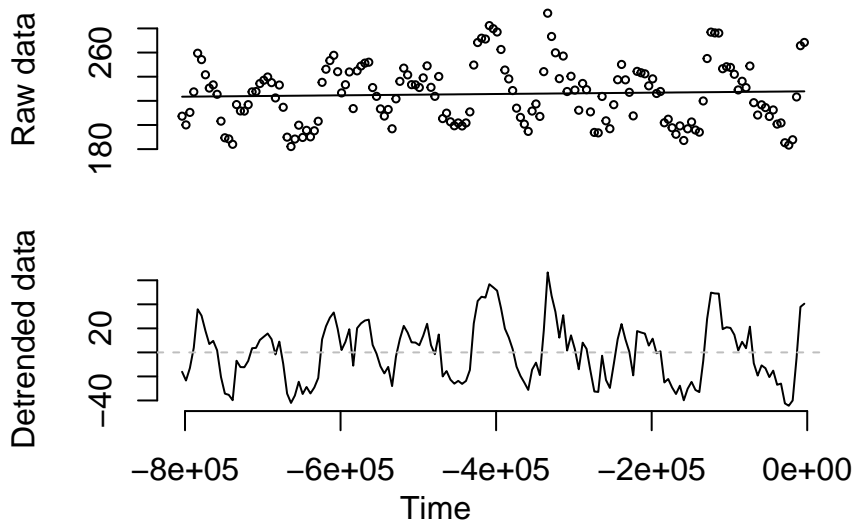
Generally shorter than the interglacials. The steepness of the transition from glacial to interglacial is somewhat variable, meaning that the time between the two states varies from transition to transition. This plays into this question because the question becomes, when during this variable speed transition do we consider the climate to be glacial vs interglacial.

```

trend <- lm( cs ~ years )      # detrend data
yhat  <- predict( trend )
cnew  <- cs - yhat

par( mfrow = c(2,1), mar = c( 2,5,1,2 ), omi = c( .7, .5, .1, .1), bty = 'n' )
plot( years, cs, cex = .5, xaxt = 'n', xlab = '', ylab = 'Raw data' )
lines( years, yhat )
plot( years, cnew, type = 'l', ylab = 'Detrended data' )
abline( h = 0, lty = 2, col = 'grey' )
mtext( 'Time', 1, outer = T )

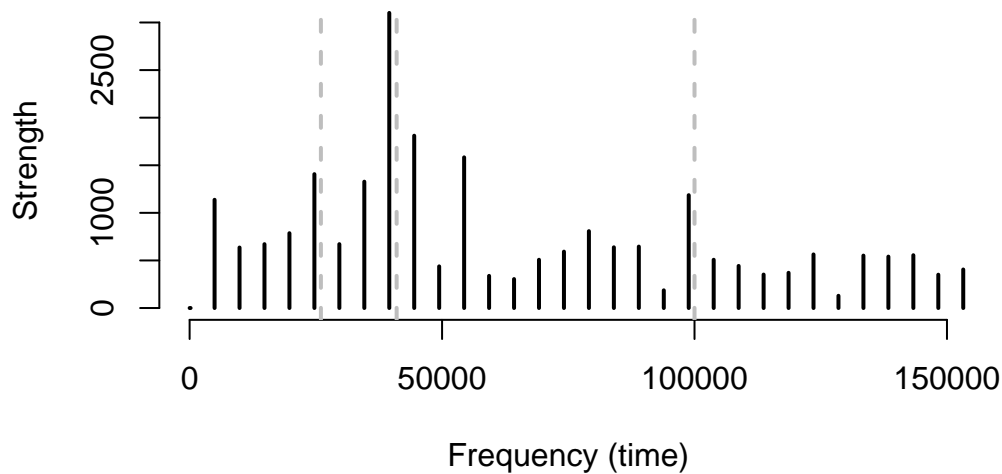
```



```

time <- diff( years )[1]
afreq <- length(cnew)/time
X.k <- fft( cnew )      # find all harmonics with fft()
milankovitch <- c( 100000, 41000, 26000 )
plotSpectrum( X.k, afreq/1000, xlim = c(0,150)*1000 )
abline( v = milankovitch, col = 'grey', lwd = 2, lty = 2 )

```



8. What are the most important frequencies in the CO2 data?

The spikes at around 25000, 32000, 52000, and 100000 seem to be the most important.

9. Convince yourself that you can precisely reproduce the trajectory if you use all frequencies ( $nh = \text{length}(X.k)$ ). Does Milankovitch appear to dominate?

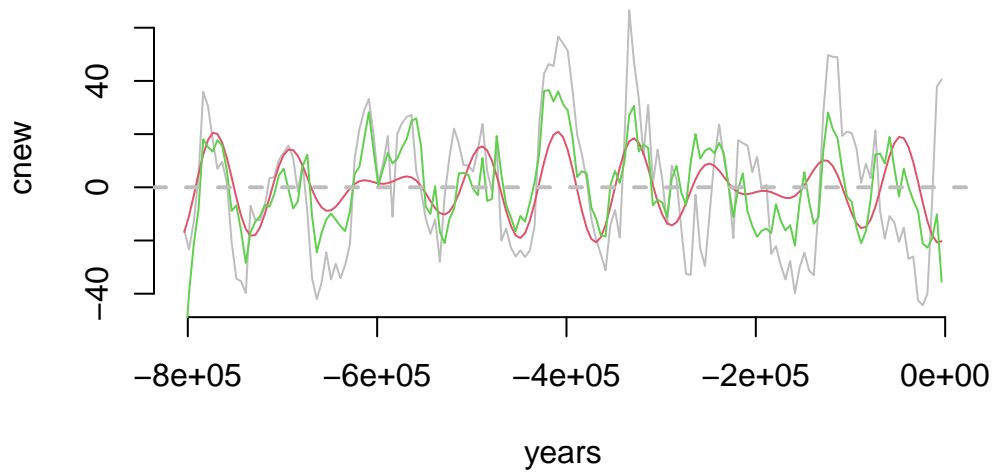
```
# Create time sequence on the sample frequency scale
ts <- seq(0, time - 1/afreq, 1/afreq) # Time sequence

# Predict trajectories
xn5 <- predictTrajectory(X.k, ts, afreq, time, nh = 4)
xn50 <- predictTrajectory(X.k, ts, afreq, time, nh = 40)
xnfull <- predictTrajectory(X.k, ts, afreq, time, nh = length( X.k ))

# Plot the original data
plot(years, cnew, type = 'l', col = 'grey', bty = 'n')

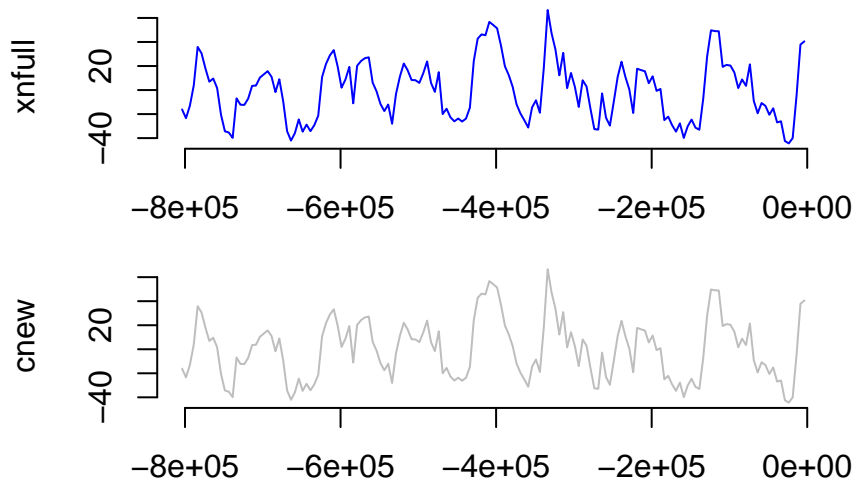
# Ensure the time values for the trajectories match the x-axis of years
# Adjust the time sequence scale to align with the 'years' axis
lines(years, xn5, col = 2)
lines(years, xn50, col = 3)

# Add a horizontal line at y = 0
abline(h = 0, lty = 2, lwd = 2, col = 'grey')
```



Here, the red line representing the four-term trajectory does seem to dominate the overall trend.

```
par( mfrow = c(2,1), mar = c( 2,5,1,2 ), omi = c( .7, .5, .1, .1), bty = 'n' )
plot(years, xnfull, type = 'l', col = 'blue', bty = 'n')
plot(years, cnew, type = 'l', col = 'grey', bty = 'n')
```



Since these plots match, we can see that the approximation is the same as the actual data in this visual-level interpretation.