EE123 Spring 2019
Discussion Section 4

Michael Chen
based on slides by Nick Antipa (sp'18)

Overlap-Save from Midterm I Spring 2018

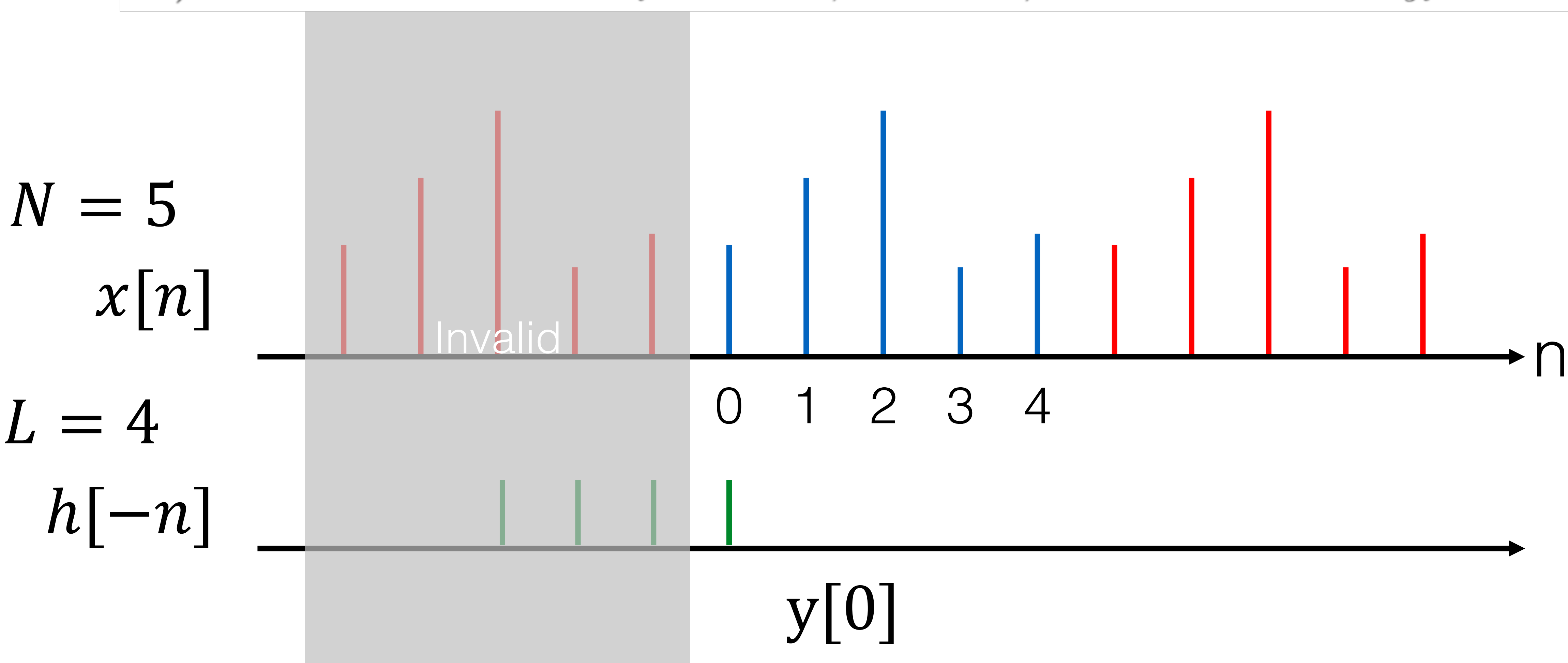
Let $h[n]$ be a **real-valued** length $L = 4$ sequence. You would like to compute the *linear* convolution

$$y[n] = x[n] * h[n],$$

using a function, which implements the overlap-and-save method.

Let $x_i[n]$ be the overlapped blocks, and let $y_i[n]$ be the saved blocks.

a) If the block-size of each x_i block is N, what is M, the block size of each y_i block?



Overlap-Save from Midterm I Spring 2018

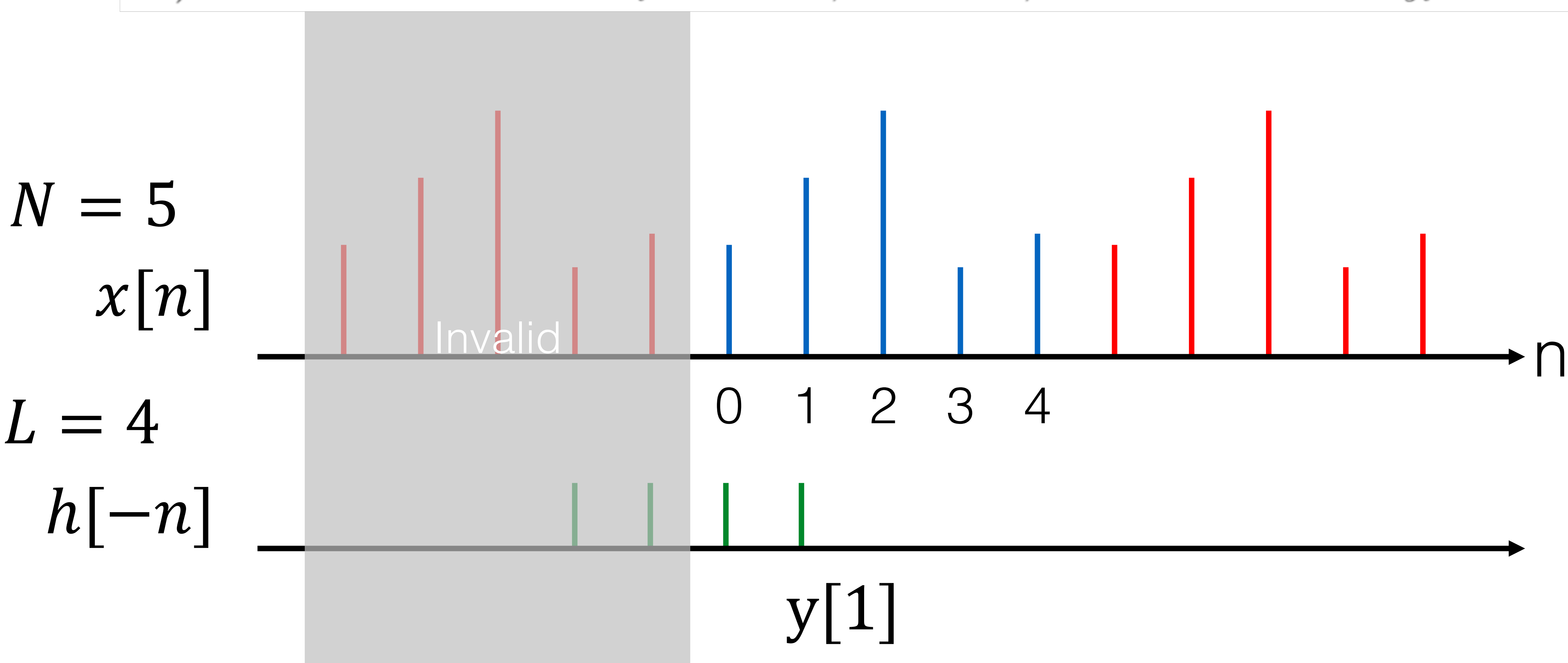
Let $h[n]$ be a **real-valued** length $L = 4$ sequence. You would like to compute the *linear* convolution

$$y[n] = x[n] * h[n],$$

using a function, which implements the overlap-and-save method.

Let $x_i[n]$ be the overlapped blocks, and let $y_i[n]$ be the saved blocks.

a) If the block-size of each x_i block is N, what is M, the block size of each y_i block?



Overlap-Save from Midterm I Spring 2018

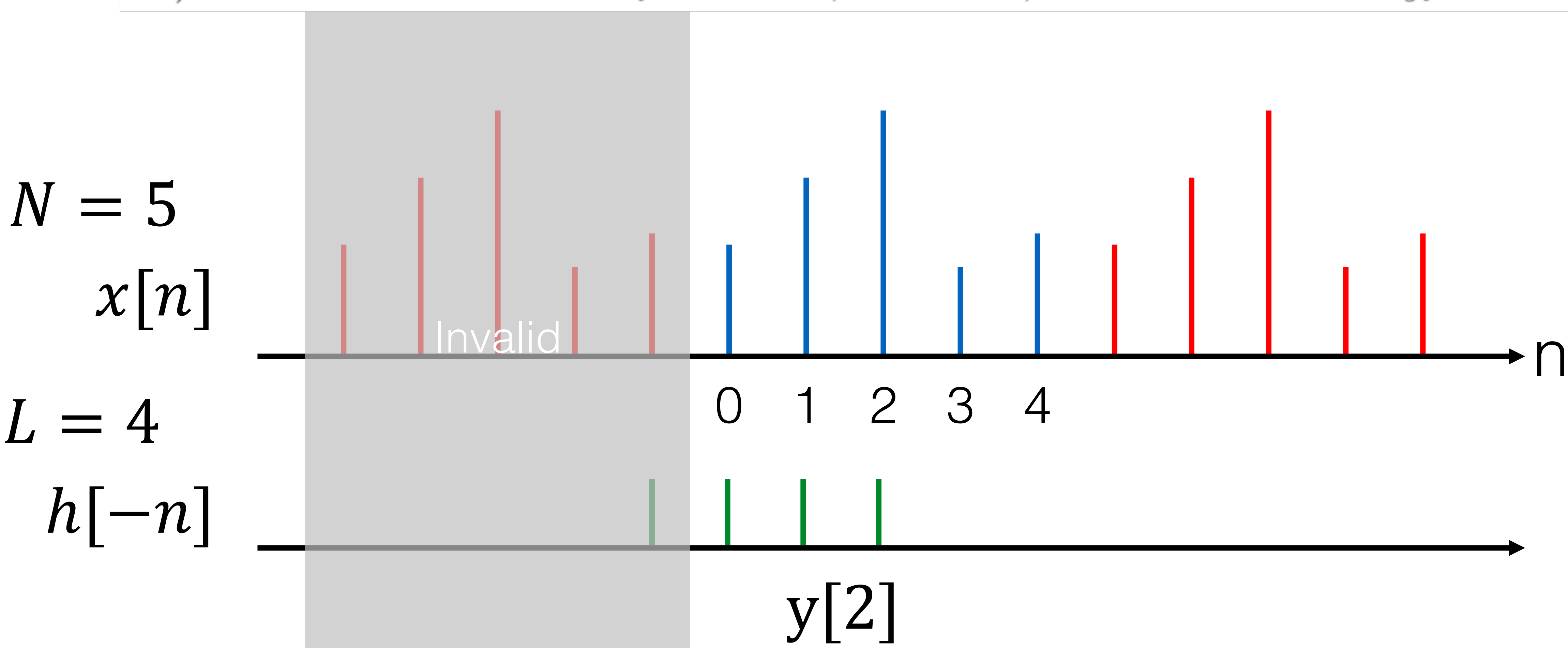
Let $h[n]$ be a **real-valued** length $L = 4$ sequence. You would like to compute the *linear* convolution

$$y[n] = x[n] * h[n],$$

using a function, which implements the overlap-and-save method.

Let $x_i[n]$ be the overlapped blocks, and let $y_i[n]$ be the saved blocks.

a) If the block-size of each x_i block is N, what is M, the block size of each y_i block?



Overlap-Save from Midterm I Spring 2018

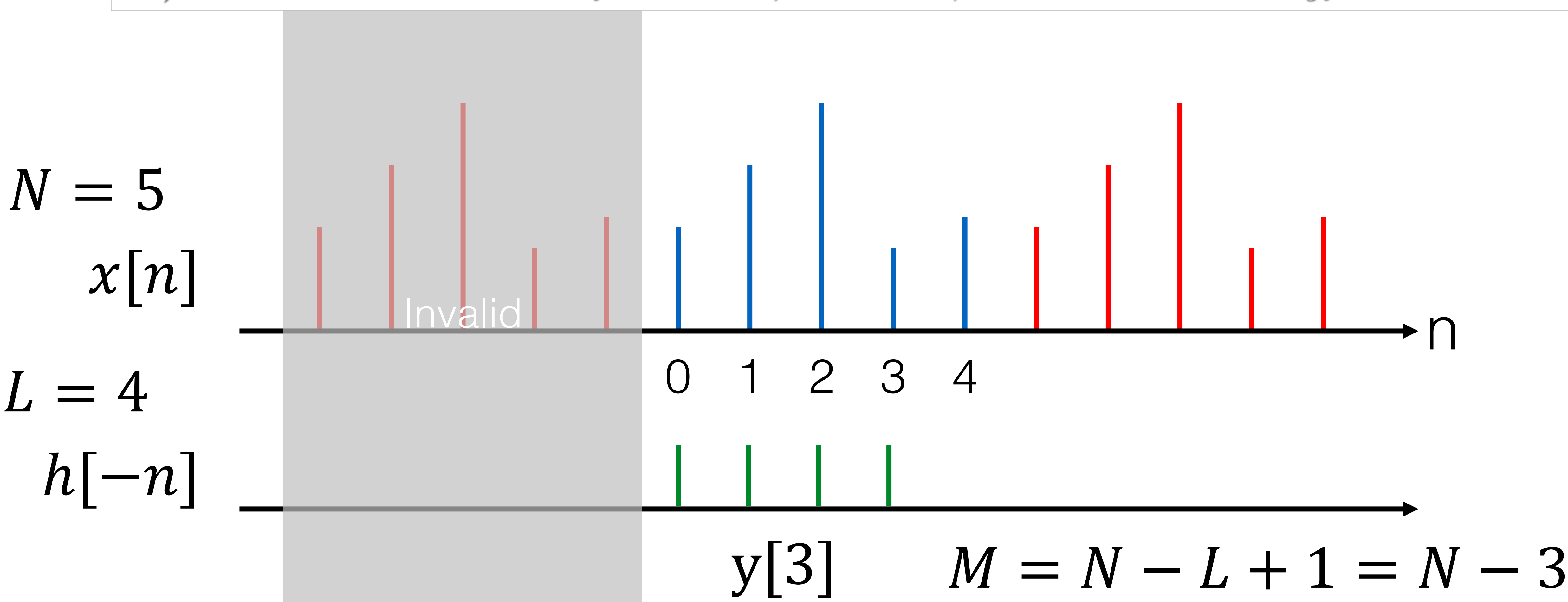
Let $h[n]$ be a **real-valued** length $L = 4$ sequence. You would like to compute the *linear* convolution

$$y[n] = x[n] * h[n],$$

using a function, which implements the overlap-and-save method.

Let $x_i[n]$ be the overlapped blocks, and let $y_i[n]$ be the saved blocks.

a) If the block-size of each x_i block is N, what is M, the block size of each y_i block?



Overlap-Save from Midterm I Spring 2018

b) You decide to use a radix-2 FFT in the overlap-and-save method. Samples of $x[n]$ are arriving at a rate of 1000 samples/sec. What is the maximum block size N you could use when the latency of your filtering system should not exceed 1 second?
(You can assume negligible time for computing O(NlogN) FFTs and O(N) multiplications)

$$N = 2^i \leq 1000 \rightarrow maximum\ block\ size\ N = 512$$

Overlap-Save from Midterm I Spring 2018

c) For computing the overlap-and-save method you precompute $H[k] = \mathrm{DFT}_N\{h[n]\}$. Unfortunately due to type-casting programming error the algorithm uses only the real-part $\tilde{H}[k] = \mathrm{Real}\{H[k]\}$ and outputs the result: $\tilde{y}[n]$. Note, that the overlap-and-save function remains unchanged and "unaware" of the bug.

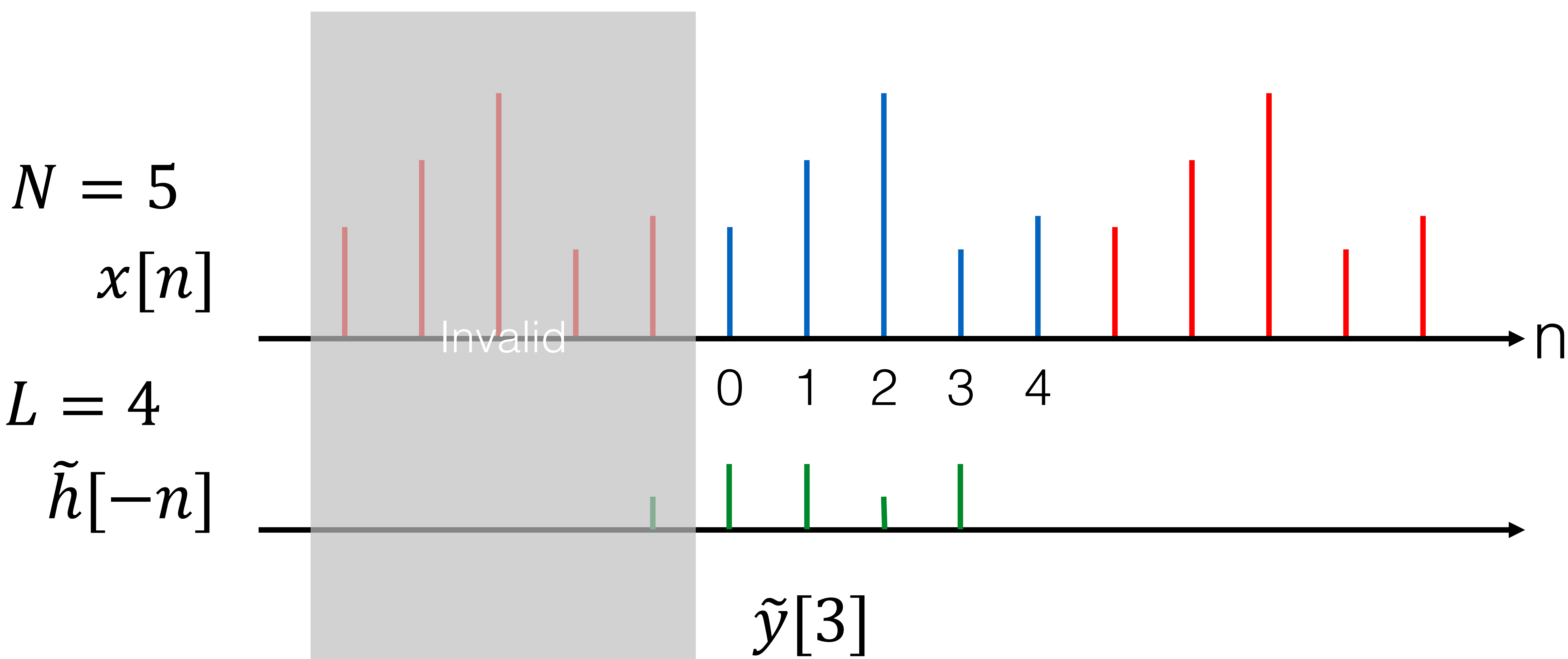
What is $\tilde{h}[n]$ the inverse DFT of $\tilde{H}[k]$? Express the solution in terms of $h_{zp}[n]$, the zero-padded original $h[n]$. (Hint: You might find symmetry properties of the DFT useful)

$$\widetilde{H}[k] = \mathrm{Real}\{H[k]\} = \frac{1}{2}(H[k] + H^*[k])$$

$$\tilde{h}[n] = \mathrm{IDFT}\left\{\frac{1}{2}(H[k] + H^*[k])\right\} = \frac{1}{2}\left(h_{zp}[n] + h_{zp}\left[\left((-n)\right)_N\right]\right)$$

Overlap-Save from Midterm I Spring 2018

d) Is the modified system in part (c) linear? Is it time-invariant?



still perform a circular convolution ➔ linear

some saved values now depends on shifted $x[n]$ ➔ not time-invariant

STFT

$$X[r,k] = \sum_{?}^{?} x[rR+m]w[m]e^{-j2\pi km/N}$$

What is the variable of summation?

STFT

$$X[r,k] = \sum_{m=0}^{?} x[rR+m]w[m]e^{-j2\pi km/N}$$

What is the variable of summation?

And the length of the sum?

STFT

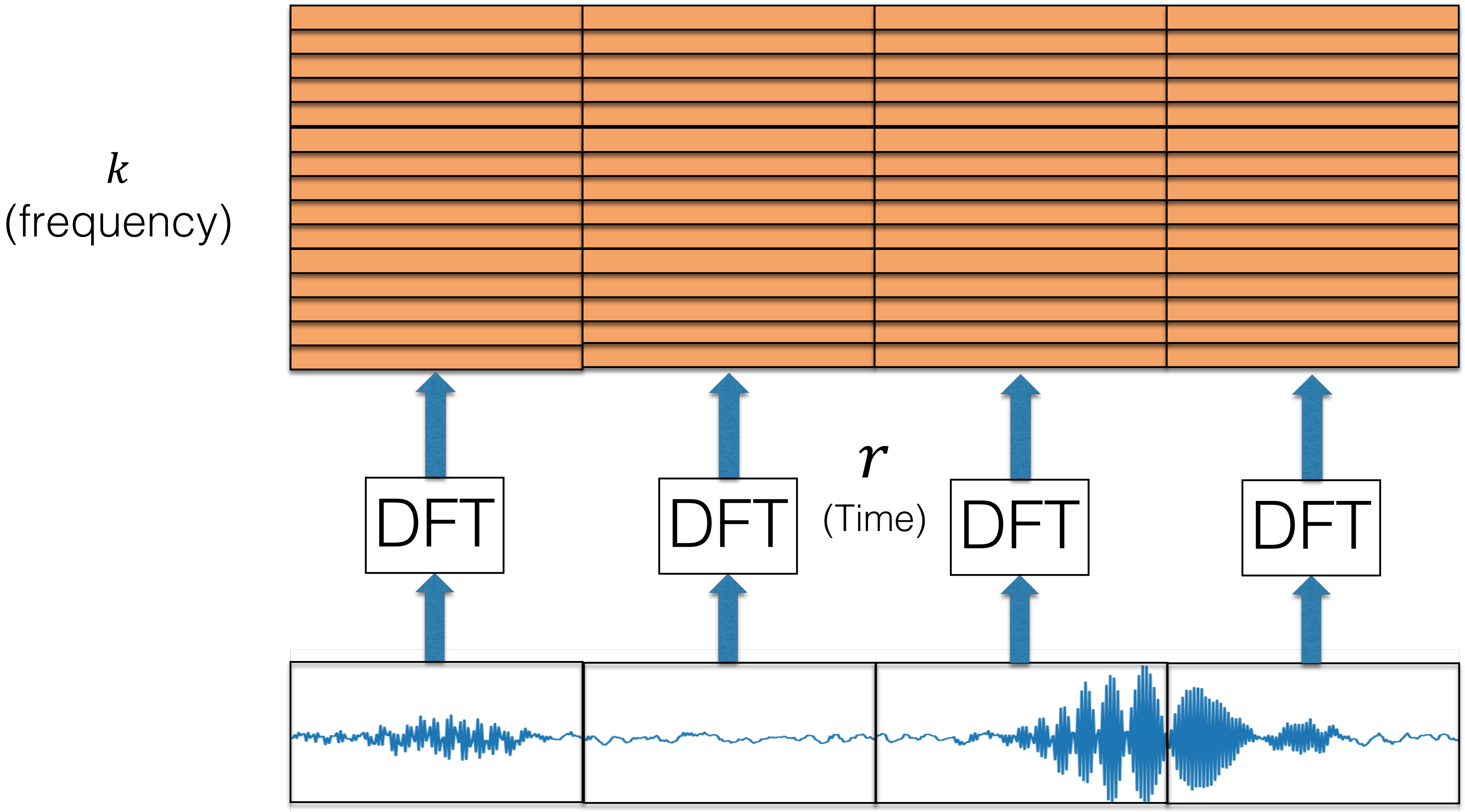
$$X[r,k] = \sum_{m=0}^{L-1} x[rR+m]w[m]e^{-j2\pi km/N}$$

What is the variable of summation?

And the length of the sum?

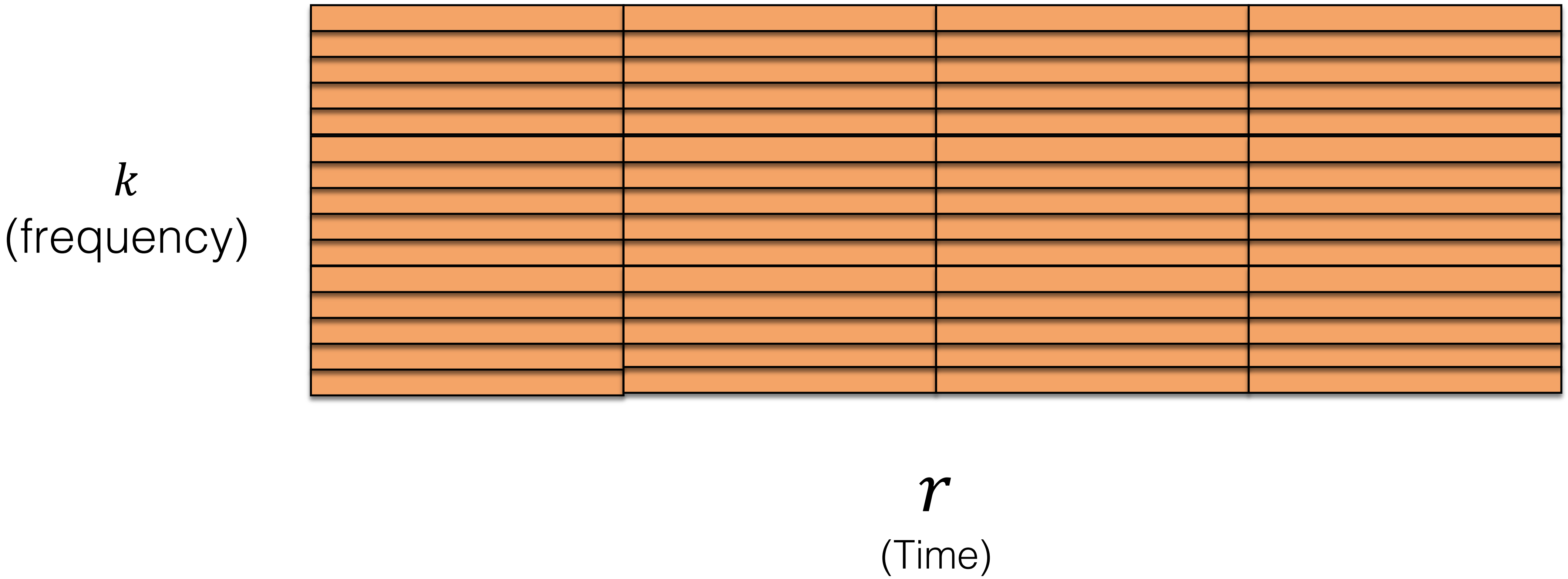
What is R for?

Non-overlapping windows



$$X[r,k] = \sum_{m=0}^{L-1} x[rR+m]w[m]e^{-j2\pi km/N}$$

Non-overlapping rect windows



$$X[r,k] = \sum_{m=0}^{L-1} x[rR+m]w[m]e^{-j2\pi km/N}$$

$$= \sum_{m=0}^{L-1} \underbrace{rect\left(\frac{m-L/2}{L}\right)} x[rR+m]e^{-j2\pi km/N}$$

What will this do to frequency space?

It will blur frequency space

Question: if I want to display a real-time equalizer of my music, what would a reasonable STFT window length be? Assume I am sampling at 48 kHz, and want to refresh my spectrum analyzer at 24 Hz.

A: 48000/24 = 2000 samples

What is my frequency resolution, in Radians and Hertz?

$$\Delta t = L = 2000$$

$$\Delta \mathrm{t}\, \Delta\omega = 2\pi$$

$$\Delta\omega = \frac{2\pi}{2000}$$

$$\Delta f = \frac{48{,}000}{2000} = 24\ Hz$$

Python demo