

Esercitazione scrittura funzionale ed operatori logici su vettori - piecewise linear interpolation

Il nostro obiettivo è quello di costruire un segnale onda pressoria arteriosa tramite l'assegnazione di istanti temporali chiave e corrispondenti valori all'interno di un intervallo temporale definito e campionato in maniera regolare. Questa operazione si chiama interpolazione a tratti in italiano o "piecewise linear interpolation" in inglese.

Opereremo come segue:

1. cerchiamo una figura di onda pressoria in rete per capire che istanti temporali chiave dovremmo poi assegnare
2. costruiamo una funzione in grado di eseguire l'interpolazione a tratti facendoci aiutare dall'intelligenza artificiale

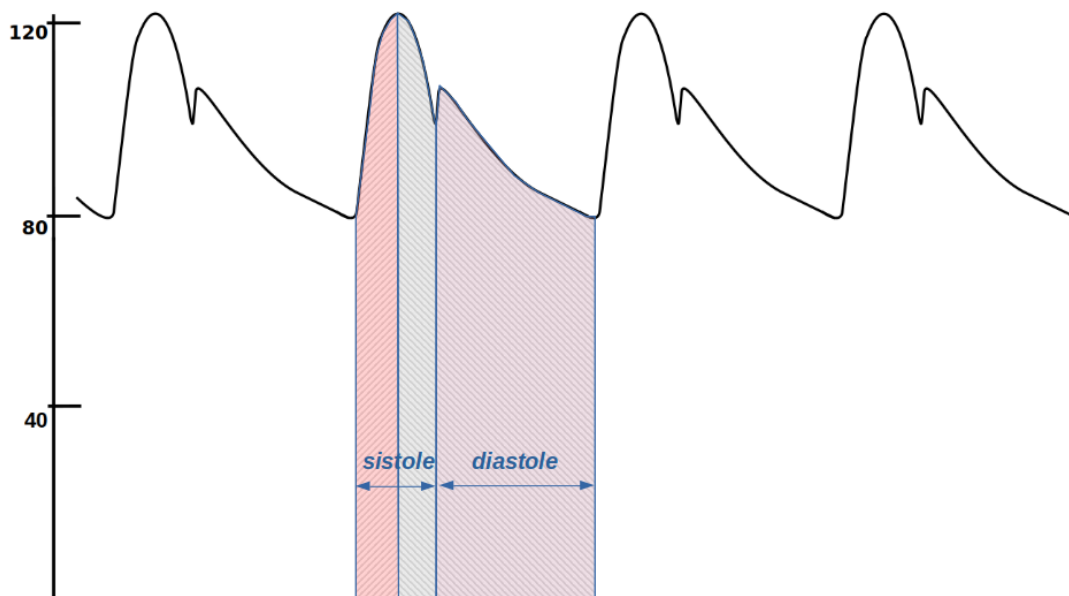


Figura – tipico andamento dell'onda pressoria arteriosa.

Supponiamo di non conoscere neppure come si chiama l'operazione che ci accingiamo a fare (interpolazione a tratti) e poniamo la seguente domanda:

DOMANDA A CHAT: “se voglio iterare mediante purr su un vettore andando a usare degli indici di un altro vettore per fare dei conti. Ad esempio, ho un vettore "t" base dei tempi da 1 a 10. Poi un vettore tind = c(0, 3, 6, 10), un vettore tval = c(0, 1, 1, 0) ed applicare tramite programmazione funzionale una interpolazione dei valori tind e tval sulla base dei tempi?”

RISPOSTA CHAT:

```
library(purrr)
```

```
t    <- 1:10          # base dei tempi
tind <- c(0, 3, 6, 10) # istanti "chiave"
```

```

tval <- c(0, 1, 1, 0)      # valori ai tempi chiave

library(purrr)

# Funzione per interpolare un singolo punto t_i
interp_fun <- function(ti, tind, tval) {
  # Trova l'intervallo [tind[k], tind[k+1]] contenente ti
  k <- max(which(tind <= ti & c(tind[-1], Inf) > ti))

  # Coordinate degli estremi
  t0 <- tind[k]
  t1 <- tind[k + 1]
  y0 <- tval[k]
  y1 <- tval[k + 1]

  # Interpolazione lineare
  y <- y0 + (ti - t0) * (y1 - y0) / (t1 - t0)
  return(y)
}

# Applica la funzione a tutti i valori di t
interp_t <- map_dbl(t, ~ interp_fun(.x, tind, tval))

# Visualizza
data.frame(t, interp_t)

```

COSA CAPISCO: che devo definire all'interno della funzione `map_dbl()` appartenente al pacchetto `purrr` una funzione che, a partire dai valori del tempo appartenenti ad un vettore (che verrà 'iterato' all'interno di `map_dbl`) trova la corrispondenza con i tratti definiti dai due vettori più piccoli (istanti e valori "chiave") e restituisce il valore interpolato.

COSA NON CAPISCO:

Mi risulta complicato capire questa linea di codice:

```

# Trova l'intervallo [tind[k], tind[k+1]] contenente ti
k <- max(which(tind <= ti & c(tind[-1], Inf) > ti))

```

Poiché anche l'AI può commettere errori e comunque è meglio produrre codice di cui si ha il pieno controllo decido di procedere autonomamente su questa parte e di riscriverla in toto.

LO RIFACCIO IN MANIERA AUTONOMA:

Allo scopo di costruire in autonomia la funzione di interpolazione a tratti da usare all'interno di `map_dbl()` occorre anzitutto trovare dove si trova l'istante **ti** nel vettore degli istanti "chiave" ed i corrispondenti valori chiave **tval** dell'intervallo, poi il compito si risolve in una semplice interpolazione lineare:

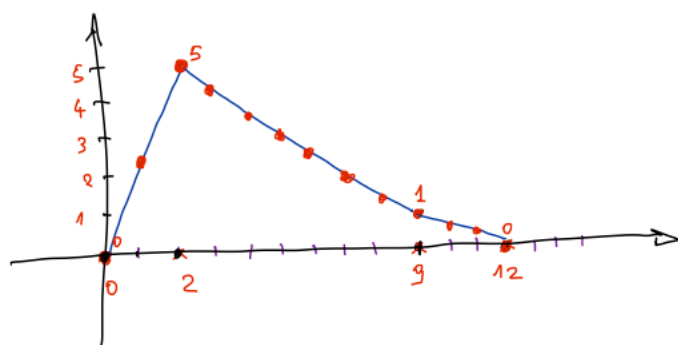
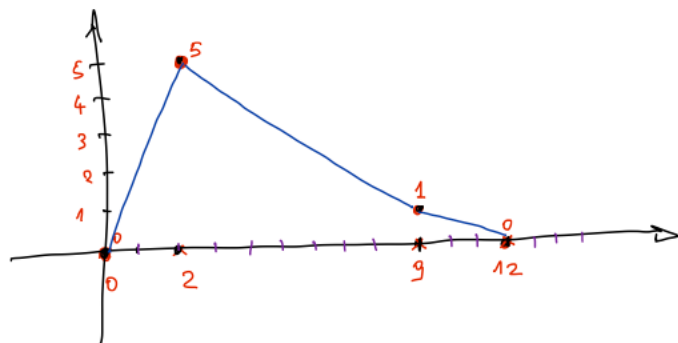
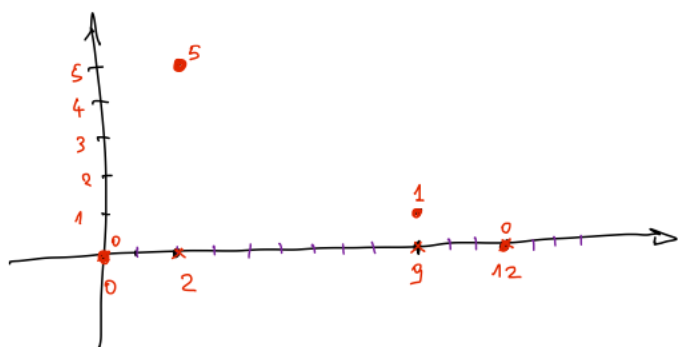
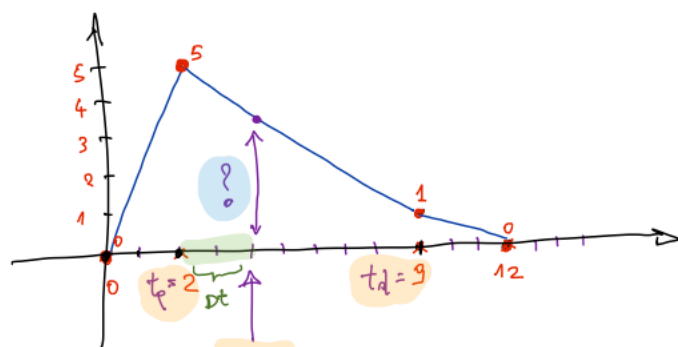


Figura – in alto sono evidenziati i soli punti chiave (tempi e valori corrispondenti). Nella figura nel mezzo è disegnata l'interpolazione lineare a tratti. Nella figura in basso il risultato finale che dobbiamo ottenere: un valore per ogni campione nella base dei tempi.



$$t_{ind} = [0 \ 2 \ 9 \ 12]$$

$$t_{val} = [0 \ 5 \ 1 \ 0]$$

1 2 3 4 ← indici ii'

$$Dt \leftarrow t_i - t_{ind} = [4 \ 2 \ -5 \ -8]$$

$$Dt [Dt \geq 0] = [4 \ 2] \quad Dt$$

$$ii \leftarrow \text{length}(Dt) = 2$$

$$P \in [0, 1] : \frac{t_{val}[ii+1] - t_{val}[ii]}{t_{ind}[ii+1] - t_{ind}[ii]} \cong P$$

$$? = t_{val}[ii] + P \cdot Dt$$

Figura – flusso logico relativo all'interpolazione della funzione lineare a tratti su un singolo istante temporale ti .

Sequenza delle operazioni:

→ Parto calcolando $Dt \leftarrow ti - tind$

Dt avrà dei valori positivi via via più piccoli poi dei valori negativi

→ Togliamo quelli negativi

DOMANDA A CHAT: “come faccio ad estrarre in R da un vettore solo gli elementi positivi o uguali a zero?”

RISPOSTA CHAT:

`x[x >= 0]`

Poi prendiamo l'ultimo che ci fornirà la differenza temporale con il valore precedente dell'intervallo e memorizziamo i valori chiave corrispondenti.

Facciamo una prova con R:

CODICE DI PROVA:

```
```{r}

t <- 1:10 # base dei tempi
tind <- c(0, 3, 6, 10) # istanti "chiave"
tval <- c(0, 0, 3, 0) # valori ai tempi chiave
ti <- t[4]

Dt <- ti - tind
Dt
Dt <- Dt[Dt>0]
Dt
ii <- length(Dt)
ii
t_p <- tind[ii]
t_p
val_p <- tval[ii]
val_p

t_d <- tind[ii+1]
t_d
val_d <- tval[ii+1]
val_d
Dt
Dt <- Dt[length(Dt)]
Dt

```
```

OUTPUT:

```
[1] 4 1 -2 -6
[1] 4 1
[1] 2
[1] 1
[1] 1
[1] 4 1
[1] 1
```

Abbiamo quindi tutto per interpolare così:

```
val_interp <- val_p + Dt * (val_d - val_p) / (t_d - t_p)
```

CODICE DEFINITIVO:

```
```{r}

t <- 1:10 # base dei tempi
tind <- c(0, 3, 6, 10) # istanti "chiave"
tval <- c(0, 0, 3, 0) # valori ai tempi chiave

interp_fun <- function(ti, tind, tval) {
 # "ti" istante da interpolare, "tind" istanti "chiave", "tval" valori ai tempi chiave
 Dt <- ti - tind
 Dt <- Dt[Dt>=0]
 ii <- length(Dt)

 Dt <- Dt[ii] # scarto con tempo chiave corrente

 t_p <- tind[ii] # valore temporale chiave precedente
 val_p <- tval[ii] # valore chiave precedente

 t_d <- tind[ii+1] # valore temporale chiave successivo
 val_d <- tval[ii+1] # valore chiave successivo

 # Interpolazione lineare e restituzione valore in uscita della funzione
 val_p + Dt * (val_d - val_p) / (t_d - t_p)
}

interp_fun(5, tind, tval)

```
```

OUTPUT:

```
[1] 2
```