moodle_2_03-06-13-54

Alan Coila Bustinza

2022-06-03

```r
library(knitr)     # For knitting document and include_graphics function
library(ggplot2)   # For plotting
library('png')
```

pregunta 1

```r
img1_path <- "p1_2022-06-03_162738.png"
include_graphics(img1_path)
```

Un río tiene una anchura de $\frac{5}{6}$m. En función de la distancia a la orilla, la profundidad del río, medida en sección recta, viene dada por la siguiente tabla:

$$\left[\begin{array}{cccccccc} Distancia & 0 & \frac{2}{9} & \frac{4}{9} & a & \frac{5}{9} & \frac{7}{9} & \frac{5}{6} \\ Profundidad & 6.438 & 6.612 & 6.778 & 6.799 & 6.859 & 7.016 & 7.054 \end{array}\right]$$

Si sabemos que el area de la sección es, aproximadamente, $5.628\ m^2$, usando la regla del trapecio compuesta dad una aproximación del valor a

Respuesta:

```r
x1 <- c(0,2/9,4/9,5/9,7/9,5/6)
y1 <- c(6.438,6.612,6.778,6.859,7.016,7.054)
polyinterp=function(x,y)
{
  if(length(x)!=length(y))
    stop ("La longitud de los vectores x e y debe ser la misma" )
  n=length(x)-1
  vandermonde=rep(1,length(x))
  for(i in 1:n)
  {
    xi=x^i
    vandermonde=cbind(vandermonde,xi)
  }
```

```r
  beta=solve(vandermonde,y, tol=1e-22)
  names(beta)=NULL
  return(beta)
}
poli <- polyinterp(x1,y1)
y_value <- 6.799
poli[1] <- poli[1]-y_value

polyroot(poli)[1]
```

```
## [1] 0.4730688+0i
```

pregunta 2

```r
img1_path <- "p2_2022-06-03_164846.png"
include_graphics(img1_path)
```

Los valores de una cierta función $f(x)$ en unos puntos concretos están recogidos en la siguiente tabla de valores:

$$\begin{bmatrix} x_i & 1 & 2 & 4 & 5 & 8 \\ f(x_i) & -\frac{7}{2} & 4 & 190 & \frac{929}{2} & 2842 \end{bmatrix}$$

Determinad el coeficiente de grado máximo del polinomio interpolador de grado 4.

```r
x2 <- c(1,2,4,5,8)
y2 <- c(-7/2,4,190,929/2,2842)
c_grado <- length(x2)
grado <- c_grado-1
polyinterp=function(x,y)
{
  if(length(x)!=length(y))
    stop ("La longitud de los vectores x e y debe ser la misma" )
  n=length(x)-1
  vandermonde=rep(1,length(x))
  for(i in 1:n)
  {
    xi=x^i
    vandermonde=cbind(vandermonde,xi)
  }
  beta=solve(vandermonde,y, tol=1e-22)
  names(beta)=NULL
  return(beta)
```

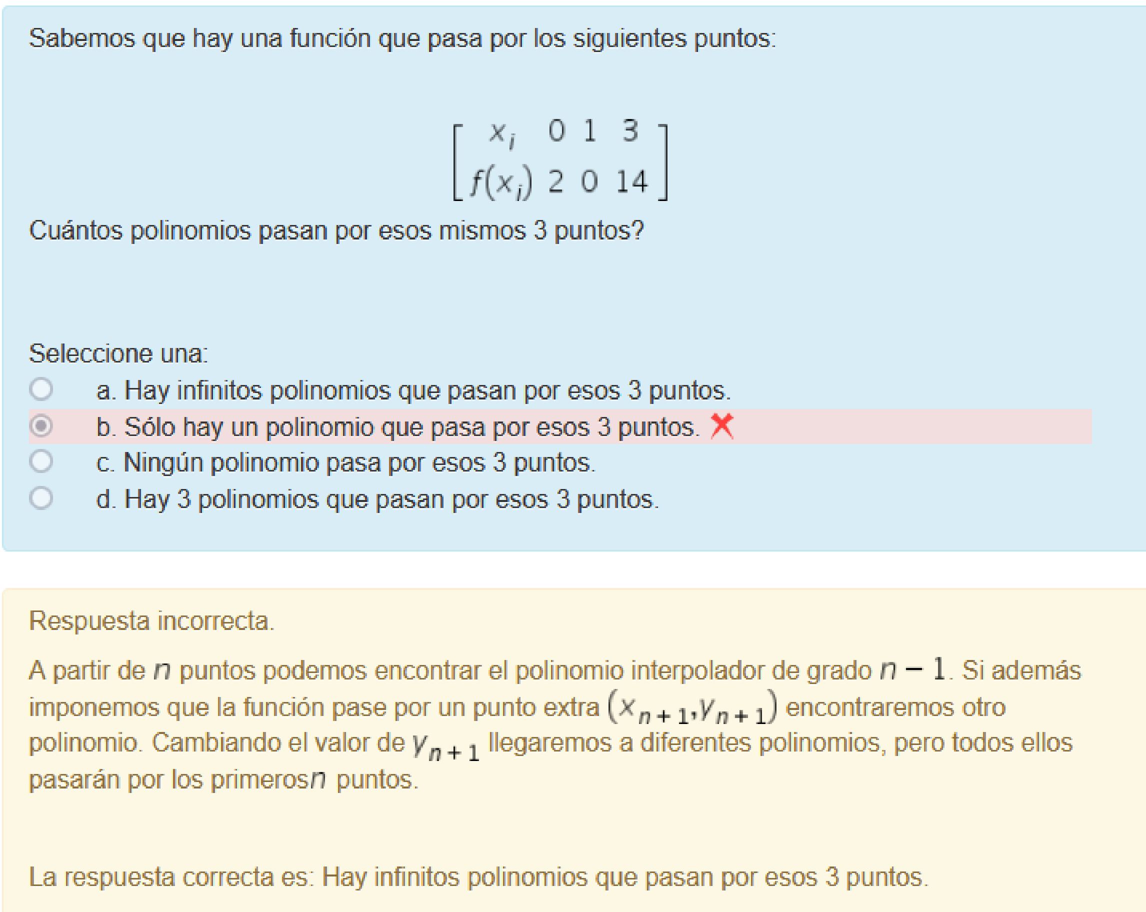
```
}

# cuidado con los coeficietnes 0 q alteran el grado maximo
polyinterp(x2,y2)[c_grado]
```

```
## [1] 0.5
```

pregunta 3

```
img1_path <- "p3_2022-06-03_165229.png"
include_graphics(img1_path)
```



Sabemos que hay una función que pasa por los siguientes puntos:

$$\begin{bmatrix} x_i & 0 & 1 & 3 \\ f(x_i) & 2 & 0 & 14 \end{bmatrix}$$

Cuántos polinomios pasan por esos mismos 3 puntos?

Seleccione una:

- a. Hay infinitos polinomios que pasan por esos 3 puntos.
- b. Sólo hay un polinomio que pasa por esos 3 puntos.
- c. Ningún polinomio pasa por esos 3 puntos.
- d. Hay 3 polinomios que pasan por esos 3 puntos.

Respuesta incorrecta.

A partir de n puntos podemos encontrar el polinomio interpolador de grado $n-1$. Si además imponemos que la función pase por un punto extra (x_{n+1}, y_{n+1}) encontraremos otro polinomio. Cambiando el valor de y_{n+1} llegaremos a diferentes polinomios, pero todos ellos pasarán por los primeros n puntos.

La respuesta correcta es: Hay infinitos polinomios que pasan por esos 3 puntos.

pregunta 4

```
img1_path <- "p4_2022-06-03_165357.png"
include_graphics(img1_path)
```

Un río tiene una anchura de $\frac{25}{36}$ m. En función de la distancia a la orilla, la profundidad del río, medida en sección recta, viene dada por la siguiente tabla:

$$\begin{bmatrix} \textit{Distancia} & 0 & \frac{1}{9} & \frac{11}{36} & \frac{4}{9} & \frac{5}{9} & \frac{23}{36} & \frac{25}{36} \\ \textit{Profundidad} & 6.931 & 7.068 & 7.3 & 7.458 & 7.582 & 7.672 & 7.732 \end{bmatrix}$$

Usando la regla del trapecio compuesta, dad una aproximación del area de la sección.

```
x4 <- c(0,1/9,11/36,4/9,5/9,23/36,25/36)
y4 <- c(6.931,7.068,7.3,7.458,7.582,7.672,7.732)


polyinterp=function(x,y){
  if(length(x)!=length(y))
    stop ("La longitud de los vectores x e y debe ser la misma" )
  n=length(x)-1
  vandermonde=rep(1,length(x))
  for(i in 1:n)
  {
    xi=x^i
    vandermonde=cbind(vandermonde,xi)
  }
  beta=solve(vandermonde,y, tol=1e-22)
  names(beta)=NULL
  return(beta)
}

myeval <- function(x, c) {
  f <- 0
  for (i in 1:length(c)) {
    f <- f + c[i] * x^(i - 1)
  }
  return(f)
}

trap=function(x,y)

{
  m <- length(x)-1
  a <- x[1]
  b <- x[m+1]
  cof <- polyinterp(x,y)
  x=seq(a,b,length.out=m+1)

  y=myeval(x,cof)
  p.area=sum((y[2:(m+1)]+y[1:m]))
  p.area=p.area*abs(b-a)/(2*m)
  return(p.area)
```

```
}

trap(x4,y4)
```

```
## [1] 5.09875
```