

Reto 2 - La Interpolación y El Clima (10 Mayo 2021)

Jenifer Medina Yopez, Kevin Andrés Garzón Ospina, Cristian Camilo Contreras Borja, Oscar Andrés Pacheco Turizo, Julián Andres Carrillo Chiquisa

Resumen— En el siguiente trabajo realizado se hace uso de diversos datos para la determinación de un instante en espacio y tiempo teniendo en cuenta diversas variables presentadas por el problema, haciendo uso del método de Interpolación o ajustando la curva. De la misma manera es necesario hacer una recapitulación de los conceptos utilizados dentro del contenido de este artículo. Para la resolución del problema va a ser esencial la presentación de diversos datos, que a su vez serán representados de forma gráfica con el fin de tener visión clara de lo presentado.

Palabras clave— estaciones, interpolación, curvatura, datos, espacio, riging, Thiessen.

I. INTRODUCCIÓN

El uso del método de interpolación es usado principalmente para poder determinar nuevos puntos haciendo uso de puntos o datos previamente presentados. Dicho esto, para la determinación de nuevos valores y haciendo uso de soluciones necesarias para la resolución de la problemática basándose en procedimientos presentados en este documento. De la misma manera, es importante decir que en el siguiente documento se muestran procedimientos de desarrollo, análisis de datos presentados por el problema y uso de estos.

II. MARCO CONCEPTUAL

Para poder desarrollar el ejercicio propuesto, es necesario mencionar los diferentes conceptos que se necesitaron para llegar a la solución del programa planteado, por lo que es necesario tener en cuenta los siguientes conceptos y algoritmos:

- **Error Relativo:** Es necesario que, para poder hacer una determinación más exacta de los valores, se hable del error relativo, el cual está determinado por el cociente entre el error absoluto y el valor exacto.[4]

III. MÉTODOS APLICADOS

Para la resolución del ejercicio, fue necesario hacer uso de diversos métodos, de los cuales usamos con el fin de ser aplicados con los datos suministrados, los métodos utilizados fueron los siguientes:

- **Interpolación Lineal:** Es la estimación con base a una recta y un intervalo cualquiera con el fin de obtener una función que cumpla con la recta mencionada anteriormente. [5]
- **Interpolación Splines:** Interpolación determinada a polinomios que específicamente presentan oscilaciones, con el fin de determinar la función con base a los puntos de las oscilaciones. [6]

IV. DATOS DADOS

A continuación, se va a hacer referencia a los datos suministrados por el problema y que se encuentran contenidos en dos archivos de tipo .csv y .xls, En el archivo de Excel, podemos encontrar la información según fecha y lugar, además de las condiciones de la zona. Los datos encontrados dentro del archivo, y que corresponden a las columnas del archivo son:

- Año.
- Día Juliano.
- Hora.
- Temperatura Interna.
- Presión Atmosférica.
- Humedad.
- Velocidad, dirección, Velocidad y dirección máximas del Viento.
- Precipitación.
- Radiación Solar Total.
- Nivel del Agua.
- Temperatura del Suelo a 10 cm, 20 cm y 50 cm.
- Humedad del Suelo.
- Flujo de Calor en el suelo.
- Datos Relacionados con panel solar utilizado (Corriente y Batería).

Además, es necesario aclarar que el documento presenta diferentes hojas, en donde se hace referencia a los datos por cada ciudad de donde se obtuvo la información.

Para el caso del segundo archivo suministrado, y que tiene extensión .csv podemos encontrar los siguientes datos:

- Coordenada en X
- Coordenada en Y
- Estaciones

V. VALIDACIÓN CRUZADA

La validación cruzada es un procedimiento de remuestreo, que se usa principalmente con el objetivo de evaluar modelos de aprendizaje automático en una muestra limitada de datos. En este caso, se usará la validación cruzada para poder dar garantía de la veracidad de las interpolaciones, se usarán los datos de las estaciones climáticas ubicadas en el archivo .xls, con estos datos se hará una interpolación spline cúbica y una interpolación lineal. Para obtener estas interpolaciones se hará uso de las funciones *Splines* y *Approx* pertenecientes a las librerías de *Splinesfun* y *Approxfun*, respectivamente.

A. Código realizado (R)

El código usado para poder dar solución a la problemática fue desarrollado en el lenguaje de programación R, la figura 1 se aprecia el diagrama de flujo muestra los pasos realizados por el código.



Fig. 1. Diagrama de flujo validación cruzada

B. Resultados

El siguiente apartado muestra las gráficas de los datos iniciales y las comparaciones con los ajustes de spline y approx realizados.

Esta primera gráfica incluye los datos iniciales de nuestro problema. Estos aparecen con una línea color negra.

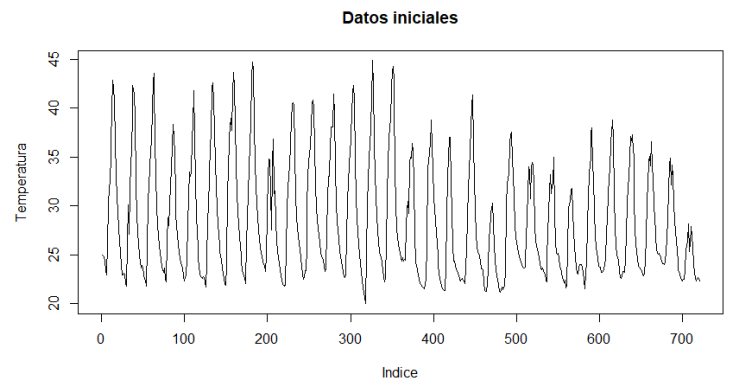


Fig. 2. Gráfica datos iniciales

Esta segunda gráfica incluye los resultados luego de aplicar el método de splines con un color *Sienna*.

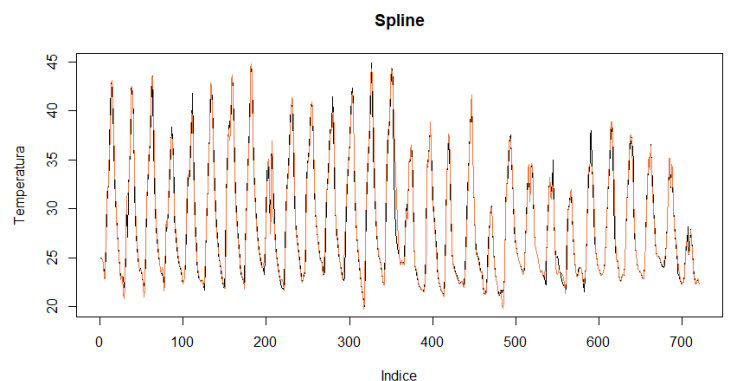


Fig. 3. Gráfica datos interpolación Spline

Esta tercera gráfica incluye los resultados luego de aplicar el método de approx con un color *Maroon*.

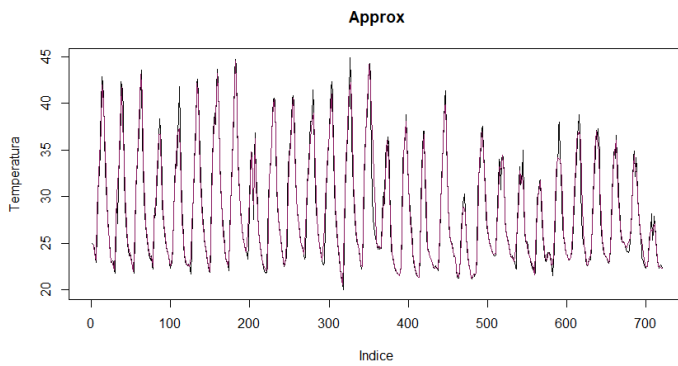


Fig. 4. Gráfica datos interpolación Aprox

La siguiente gráfica incluye la combinación del método de splines y approx. Este con un color *Sienna* para splines y un color *Maroon* para approx.

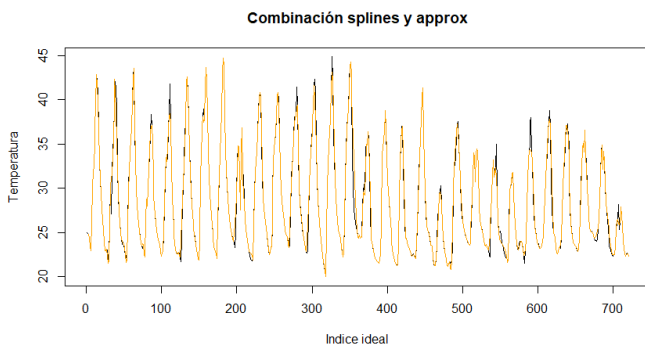


Fig. 6. Gráfica suma splines y approx

	Interpolación Spline	Interpolación Approx	Combinación Spline y Approx
Errores	195	198	193
Error mínimo	0.01	0.01	0.01
Error medio	0.03	0.03	0.03
Error máximo	0.28	0.22	0.19
Índice Jaccard	0.7292	0.725	0.7319

De la siguiente tabla podemos deducir que, aunque tenemos números de errores considerables, esto debido que al reducir el 35% de los datos de forma aleatoria para interpolar, el error se subió de manera considerable, sin embargo, el error se mantiene en valores bajos. También logramos notar que, al combinar los dos métodos, el error disminuía de manera

considerable, hasta en un 33% respecto a splines y un 24% respecto a approx lo cual significan valores muy positivos para este método.

VI. APROXIMACIÓN DE ESTACIONES

Se busca realizar una proyección de datos basadas en la estación Santa Quiteria con respecto a las demás estaciones, teniendo en cuenta los datos suministrados de tipo csv con las coordenadas de cada una de las estaciones y de la misma manera tomando en cuenta esta estación para así predecir la variable necesaria que es la temperatura.

C. Código realizado (Python/R)

El código se dividió en dos partes, la primera de ellas se realizó en Python en donde se extrajeron las coordenadas de cada estación de manera csv para así poder calcular las distancias de cada una respecto a la estación escogida, también se procedió a crear una gráfica con las ubicaciones geográficas de cada una como se indica en la figura [x], para poder encontrar la estación más cercana se realizaron diferentes bucles de comparación para así obtener los datos de cada una de las distancia y próximamente escoger la menos de estas.

```

1  # -*- coding: utf-8 -*-
2  """
3  Reto 2 - AnalisisNumerico
4  """
5
6  import pandas as pd
7  import numpy as np
8  import matplotlib.pyplot as plt
9
10 #Variables
11 nombres = []
12 cont = 0
13 y1 = 0
14 x1 = 0
15 distancias = []
16 ciudad = ""
17
18 coordenadas = pd.read_csv("coordenadas.csv")
19 print(coordenadas)
20
21 for i in coordenadas.iloc[:,2]:
22     nombres.append(i)
23
24 marco = plt.figure(figsize=(10, 15))
25 ax = plt.subplot(2, 1, 1)
26
27 #Graficas
28 for x, y in zip(coordenadas.iloc[:,0], coordenadas.iloc[:,1]):
29     plt.plot((x,), (y,), 'bo')
30     plt.text(x, y, nombres[cont])
31     cont = cont + 1
32 plt.xlabel("Coordenada X")
33 plt.ylabel("Coordenada Y")
34 plt.title("Coordenadas geograficas")
35 plt.savefig("Coordenadas geograficas.pdf")
36
37 #Encontrar la estación más cercana a Santa Quiteria
38
39 for i in range(0, len(coordenadas)):
40     if(coordenadas.iloc[i,2] == "Santa Quiteria"):
41         x1 = coordenadas.iloc[i,0]
42         y1 = coordenadas.iloc[i,1]
43
44 a = np.array([x1,y1])
45 for i in range(0, len(coordenadas)):
46     if(coordenadas.iloc[i,2] != "Santa Quiteria"):
47         x2 = coordenadas.iloc[i,0]
48         y2 = coordenadas.iloc[i,1]
49         b = ([x2, y2])
50         distancias.append((coordenadas.iloc[i,2], np.sqrt(np.sum(np.power(b-a, 2)))))
51
52 print("\n Distancias: ")
53 print(distancias)
54
55 menor = distancias[0][1]
56 for i in range(1, len(distancias)):
57     if distancias[i][1] < menor:
58         menor = distancias[i][1]
59         ciudad = distancias[i][0]
60
61 print("\nLa estación mas cerca a Santa Quiteria es la estación " + ciudad)
62

```

Fig. 7. Implementación parte1 distancias(pyhton)

Mientras que en la segunda parte del código se tomó en otro lenguaje debido al uso de las funciones necesarias para hacer las comparaciones de la temperatura interna de las aproximaciones de las estaciones también graficando el resultado de la interpolación después de la aplicación y antes con los datos dados.

```

1 library(pracma)
2 library(readxl)
3
4 arch1 = read_excel("Users/kevin andres/Desktop/Abril 2013 (1 em 1 hora).xls", sheet = "Itatira")
5 arch2 = read_excel("Users/kevin andres/Desktop/Abril 2013 (1 em 1 hora).xls", sheet = "Santa Quitéria")
6
7 x = seq(from = 1, to = 720, by = 1)
8 y = arch1$Temp. Interna ('C')
9 dias = arch1$Dia Juliano
10 horas = arch1$Hora
11 indicesIdeales = x
12 #5% de los datos
13 ones = rep(1, 720)
14 reduccion = sample.int(720, 720*0.35)
15 for (e in reduccion) {
16   ones[e] = 0
17 }
18 x2 = c()
19 y2 = c()
20 i = 1
21 j = 1
22 for (o in ones) {
23   if (o == 1) {
24     x2[i] = x[j]
25     y2[i] = y[j]
26     i = i + 1
27     j = j + 1
28   }
29 }
30
31 #Graficas
32 plot(x,y,type='l', ylab = "Temperatura interna", xlab = "Indice Ideal", col= "red")
33 lines(spline(x2,y2,n=200),col= "blue")
34 interpolacion = splinefun(x2,y2)
35 interpolados = c()
36 k = 1
37 error = c()
38 for (var in x) {
39   errorY = interpolacion(var)
40   error = c(error, abs((y[k] - errorY)/y[k]))
41   k = k + 1
42 }
43 print(error)
44 interpolados2 = c()
45 for (i in 1:length(arch2$Dia Juliano)) {
46   dia = arch2$Dia Juliano[i]
47   hora = arch2$Hora[i]
48   for (j in 1:720) {
49     if ((dias[j] == dia) && (horas[j] == hora)) {
50       interpolados2 = c(interpolados2, indicesIdeales[j])
51     }
52   }
53 }
54 nuevoY = c()
55 errorEstacion = c()
56 z = 1
57 for (variable in interpolados2) {
58   nuevoY = c(nuevoY, interpolacion(variable))
59   errorEstacion = c(errorEstacion, abs((arch2$Temp. Interna ('C')[z] - nuevoY[z])/arch2$Temp. Interna ('C')[z]))
60   z = z + 1
61 }
62 plot(interpolados2, arch2$Temp. Interna ('C'), ylab = "Temperatura interna", xlab = "Indice Calculado", type = 'l', col = "red")
63 print(errorEstacion)
64 maximo = 0
65 media = 0
66 for (error in errorEstacion) {
67   if (error > maximo) {
68     maximo = error
69   }
70   media = media + error
71 }

```

Fig. 8. Implementación parte2 aproximación e interpolación(R)

D. Gráficas y Tablas

Estaciones	Distancia (Santa Quitéria)
Vicosa do Ceara	1.215315240505757
Sobral	0.7000080839500633
Sao Goncalo do Amarante	1.3425411868551271
Santa Quitéria	2.880383881250888

Santana do Cariri	1.2261613350202576
Quixeramobim Quixada	1.2840244879343166
Pentecoste	1.0290357841275968
Jati	3.542001878073485
Itatira	0.5500478738580565
Funceme	1.6930797335157097
Crato	2.997444863589206
Camocim	1.5926572551805873
Acarau	1.4473891891069708
Araripe	2.8668921723072014
Aiuaba	2.2342504883446397
Beberibe	2.0278747257196024

TABLA I (DISTANCIA ESTACIONES CON RESPECTO A SANTA QUITERIA)

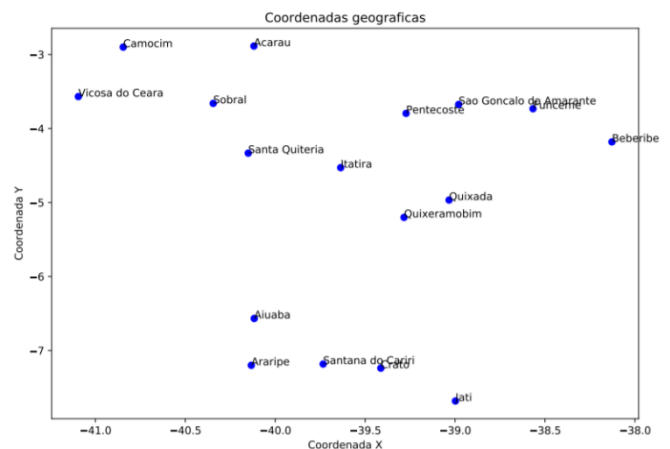


Fig. 9. Gráficas coordenadas de las estaciones

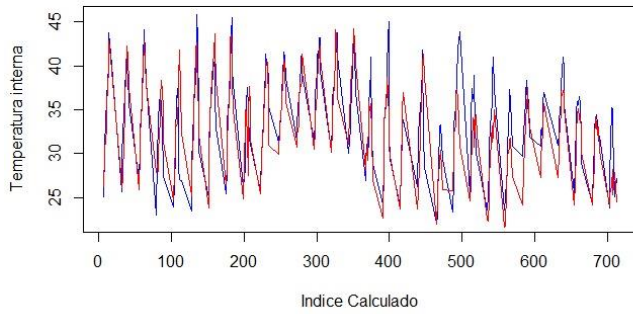


Fig.10. Dato dados vs Después de eliminación

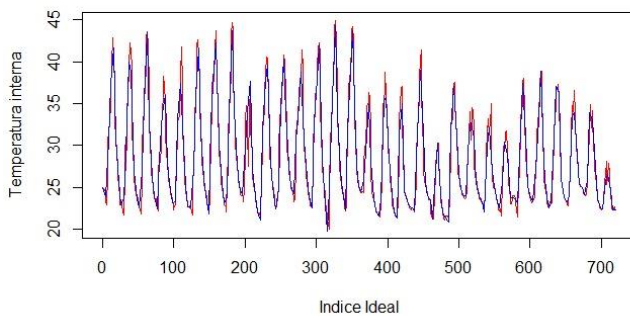


Fig. 11. Datos Dados vs Interpolación

VI. Conclusiones

Haciendo uso de los métodos mencionados al principio del documento, es posible determinar que la forma en la que es posible tener un punto medio con respecto al error de la operación, es muy útil hacer un promedio entre datos obtenidos por medio de los métodos utilizados, aun así, con todo esto es importante recalcar que el error generado por la operación se incrementó, pero a pesar de esto, sigue tomando un valor bajo.

REFERENCIAS

- [1] https://www.w3schools.com/python/pandas/pandas_csv.asp
- [2] https://matplotlib.org/2.1.1/api/_as_gen/matplotlib.pyplot.plot.
- [3] J. Fernandez and G. Coronado, "Errores Absolutos y Relativos", Fisicalab.com. [Online]. Available: <https://www.fisicalab.com/apartado/errores-absolutos-relativos>. [Accessed: 13- May- 2021].
- [4] "Interpolación", Sc.ehu.es. [Online]. Available: <http://www.sc.ehu.es/sbweb/fisica3/datos/regresion/interpolacion.html>. [Accessed: 13- May- 2021].
- [5] Ugr.es. [Online]. Available: <http://www.ugr.es/~prodelas/ftp/Ciencias/Geologia/tema3bis.pdf>. [Accessed: 13- May- 2021].
- [6] Ugr.es. [Online]. Available: <http://www.ugr.es/~prodelas/ftp/Ciencias/Geologia/tema3bis.pdf>. [Accessed: 13- May- 2021].