Nicolás Cervantes

Punto 1.

a. Para sumar los elementos de la sub matriz triangular superior o inferior, dada la matriz cuadrada An, se usó el siguiente código:

```
import
matplotlib.pyplot
as plt
```

```
def sumaTriangular(A):
    matrix = list(list(A))
    n = len(matrix)
    suma = 0
    data=[]
    n = 0
    for x in range(0,n):
        for y in range(x,n):
            suma+=matrix[x][y]
            n += 1
            data.append(suma)
    plt.plot(data)
    plt.ylabel('Convergencia')
    plt.show()
    return suma


#Ejemplo
matrizA = [[1,2,3,5],[0,1,8,7],[0,0,8,5],[0,1,7,3]]
sumaTriangular(matrizA)
```

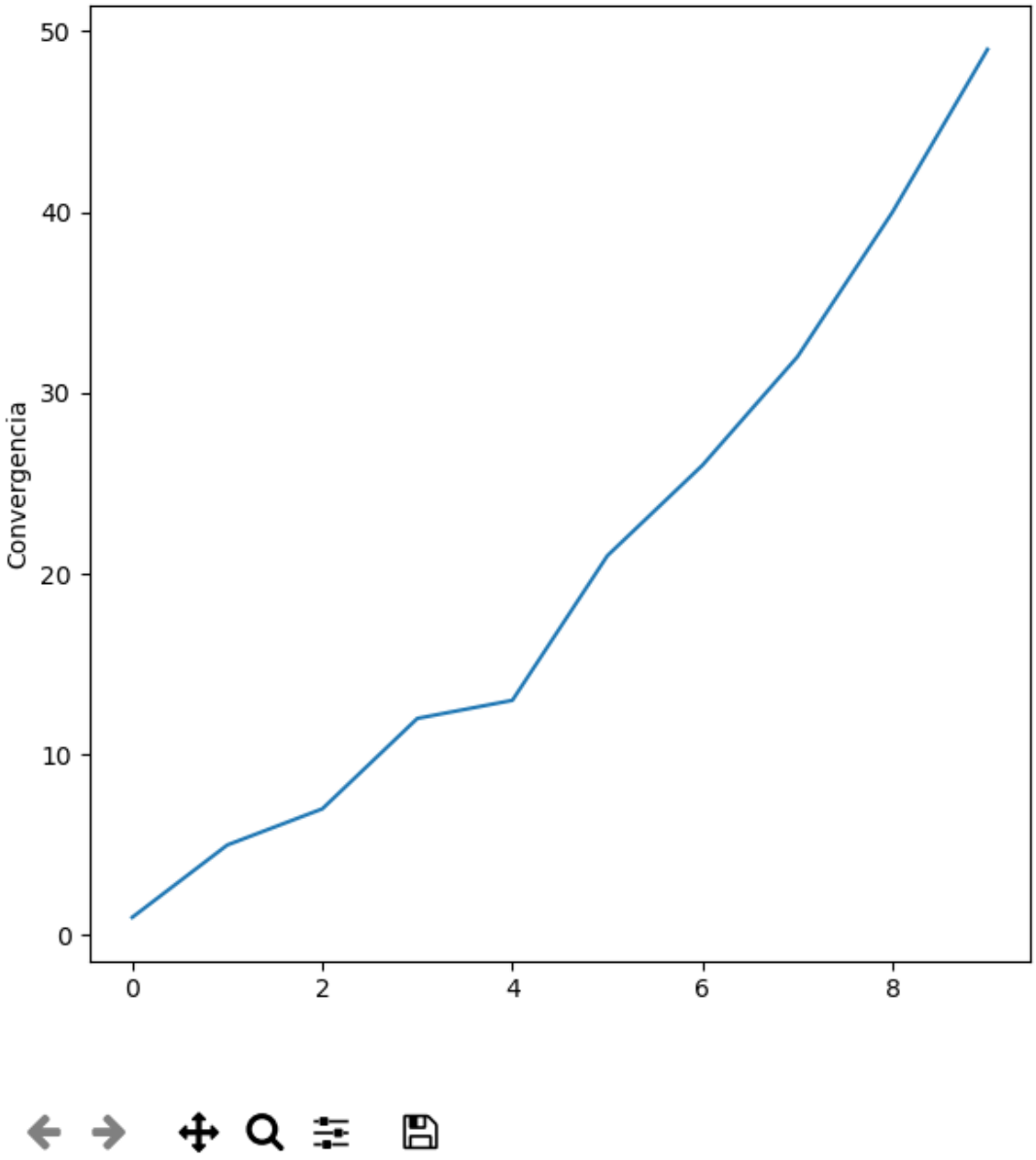
En donde el orden de convergencia es O(n^2), lo cual se puede representar bajo la siguiente gráfica

Valor de la suma:

[1, 5, 7, 12, 13, 21, 26, 32, 40, 49]

Número de operaciones:

10

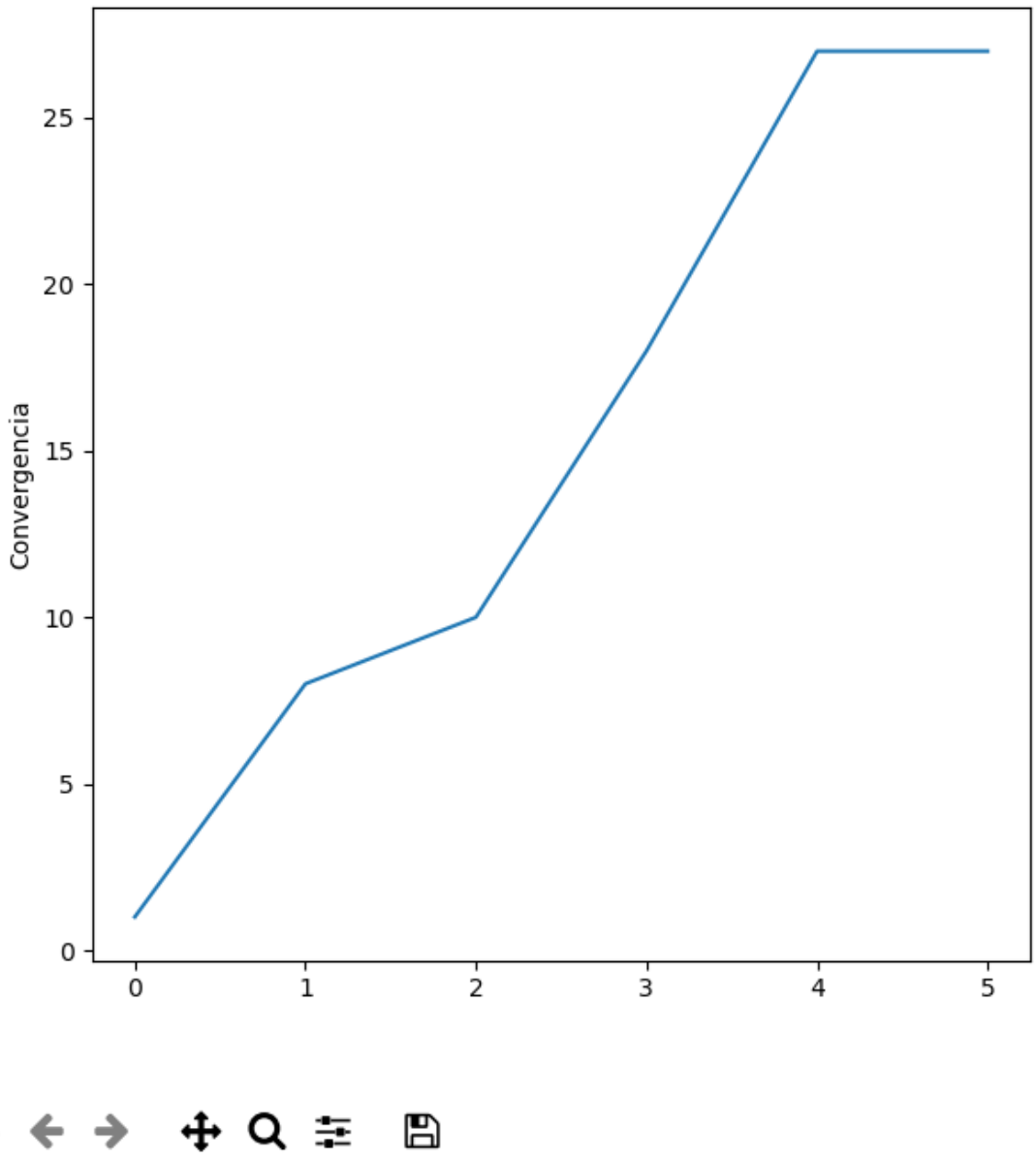


Valor de la suma:

[1, 8, 10, 18, 27, 27]

Número de operaciones:

6

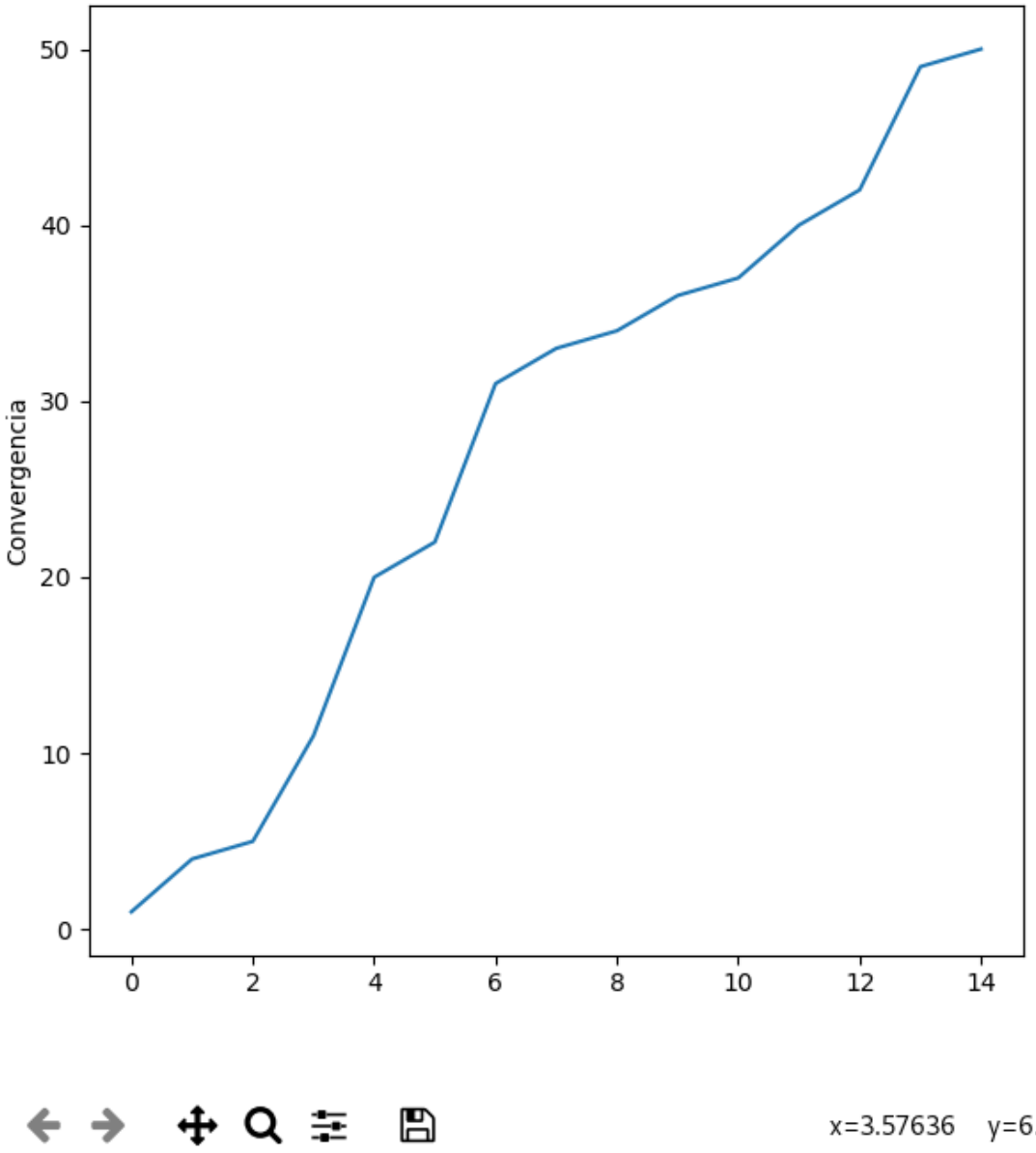


Valor de la suma:

[1, 4, 5, 11, 20, 22, 31, 33, 34, 36, 37, 40, 42, 49, 50]

Número de operaciones:

15



b. Para sumar los elementos de una matriz cuadrada An, se usó el siguiente código:

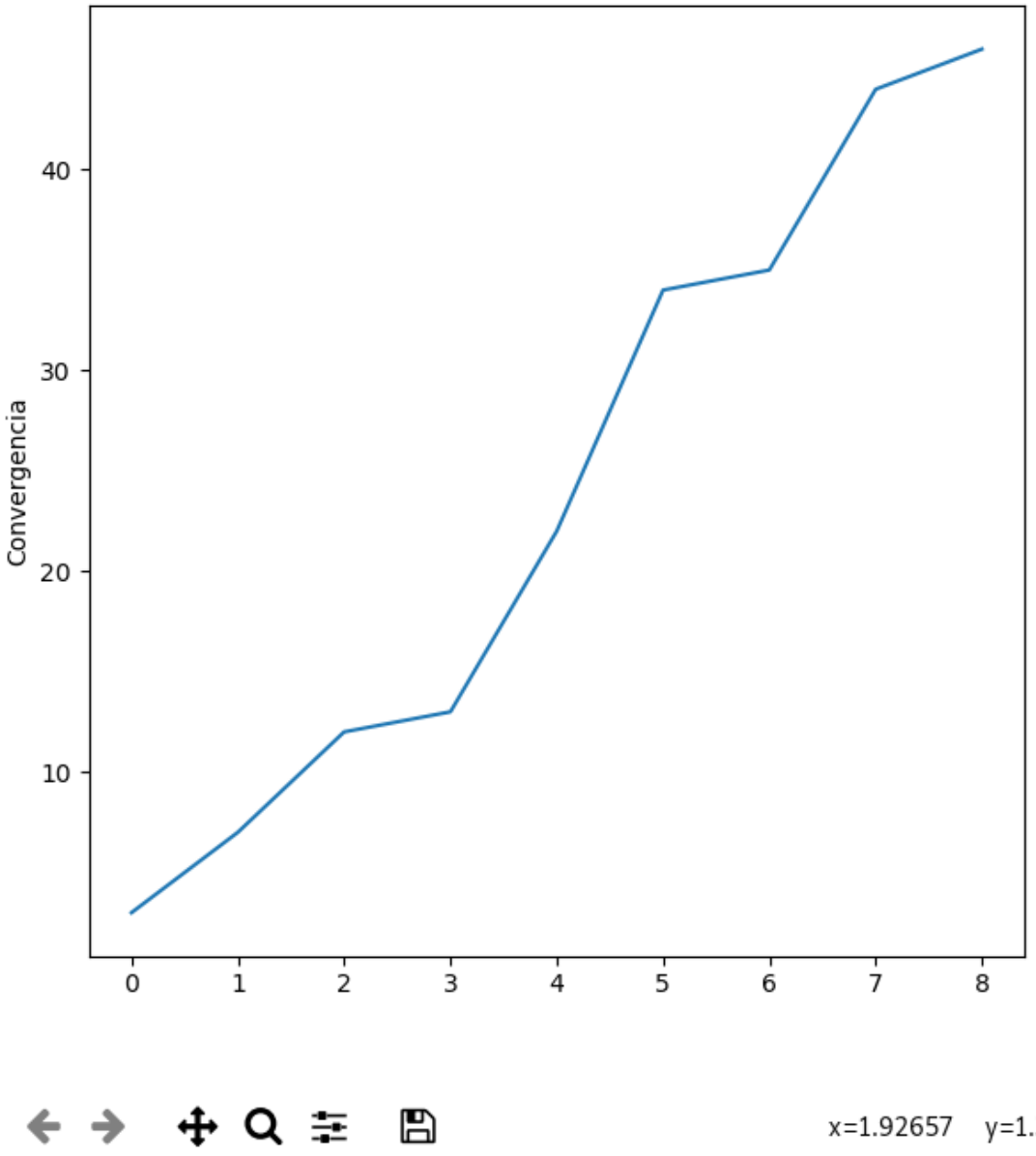
```
import
matplotlib.pyplot
as plt
```

```
def sumaElementos(x, n):
  suma=0
  data=[]
  for i in range(n):
    for j in range(n):
      suma+=x[i][j]
      data.append(suma)
  plt.plot(data)
  plt.ylabel('Convergencia')
  plt.show()

#Ejemplo

A=[[3,4,5],[1,9,12],[1,9,2]]
sumaElementos(A,3)
```

En donde el orden de convergencia se puede representar bajo la siguiente gráfica

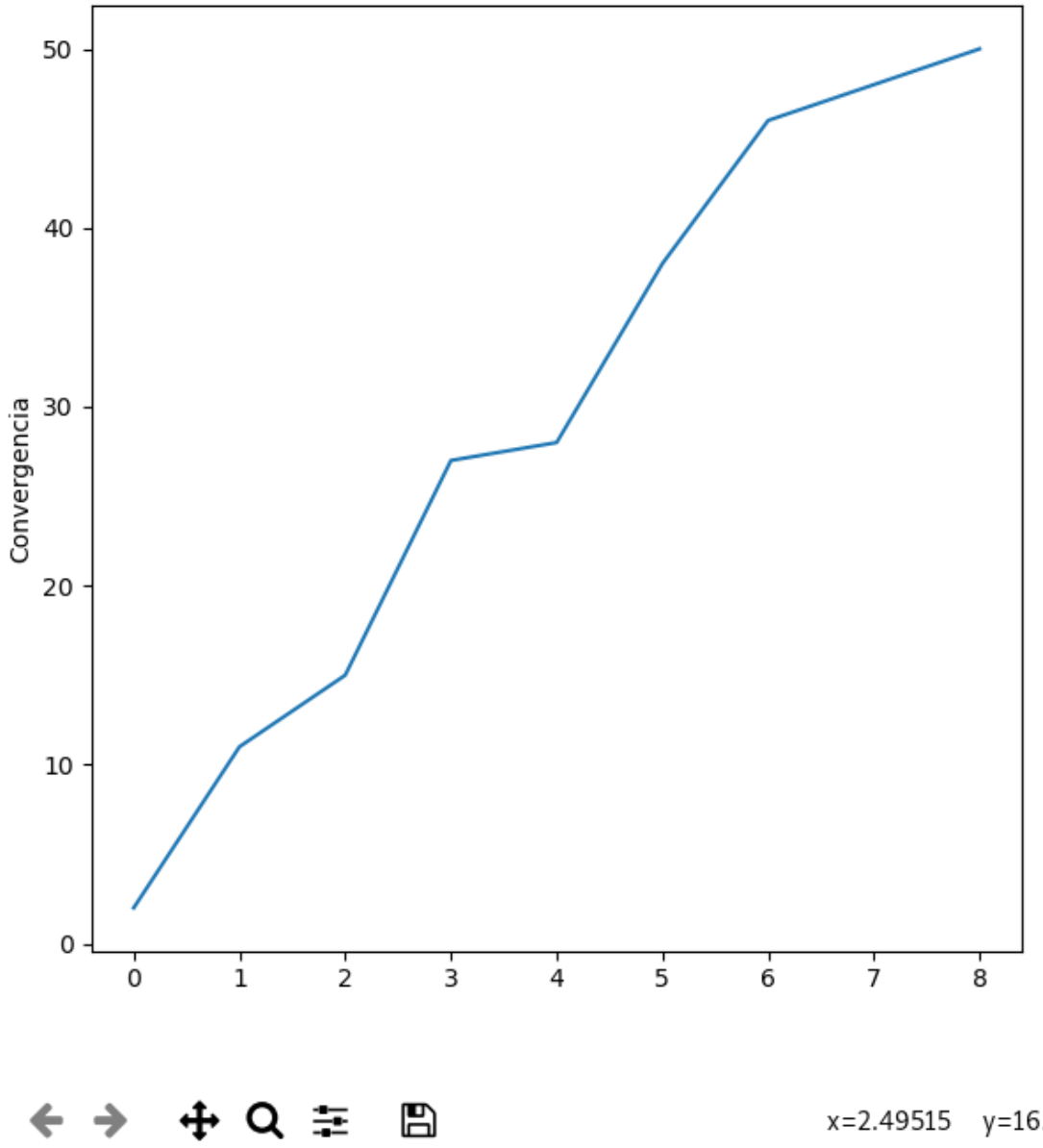
Matriz: A=[[3,4,5],[1,9,12],[1,9,2]]

Suma: 46



Matriz: A=[[2,9,4,1],[12,1,10,9],[8,2,2,2],[7,2,1,3]]

Suma: 50



x=2.49515 y=16.9563

c. Para sumar los n^2 primeros números naturales al cuadrado, se usó este código:

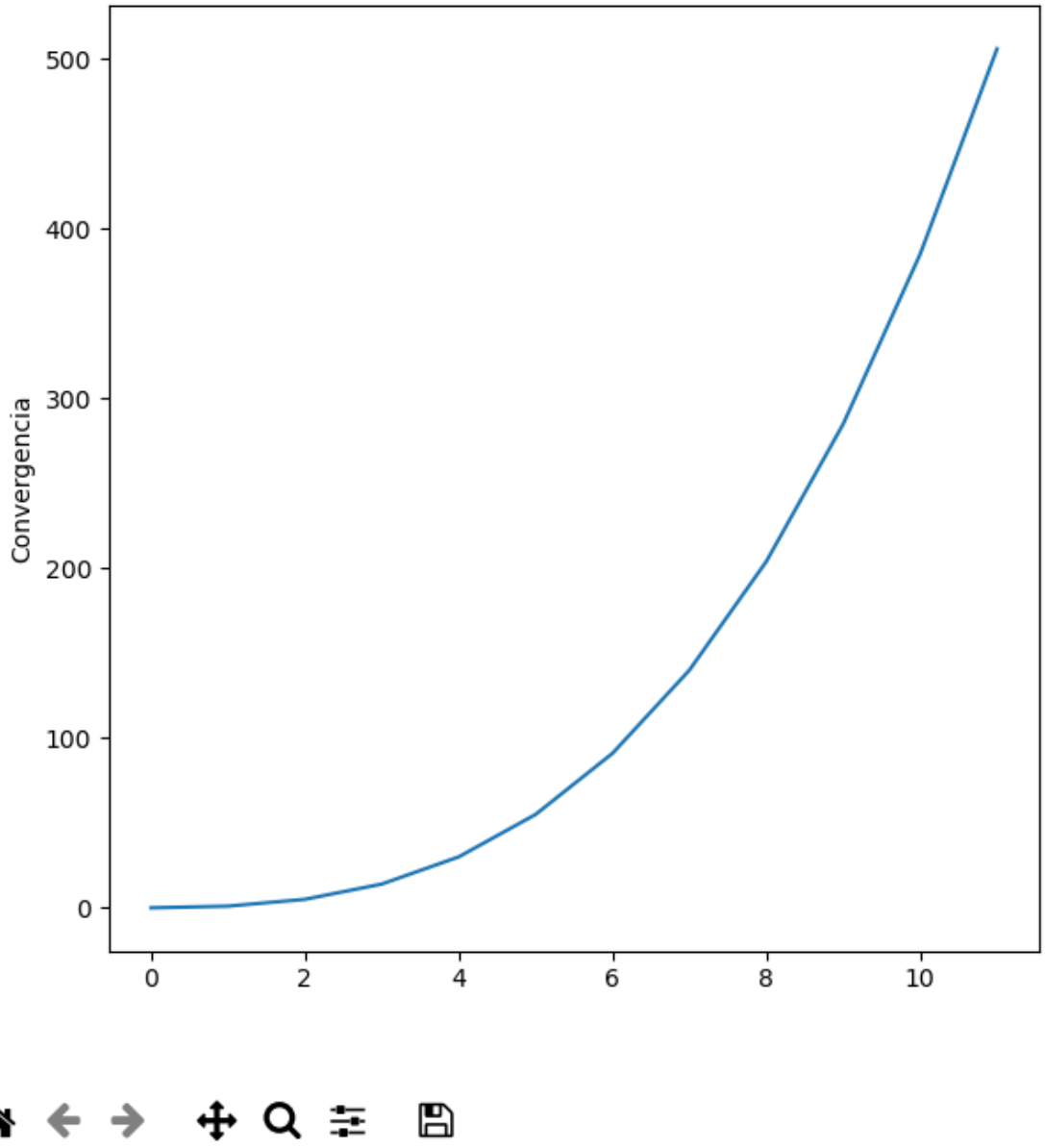
```
import
matplotlib.pyplot
as plt


                    def primerosCuadrados(n):
```

```
        suma=0
        data=[]
        for i in range(n):
                suma+=pow(i,2)
                data.append(suma)
        plt.plot(data)
        plt.ylabel('Convergencia')
        plt.show()


primerosCuadrados(5)
```

En donde el orden de convergencia se puede representar bajo la siguiente gráfica

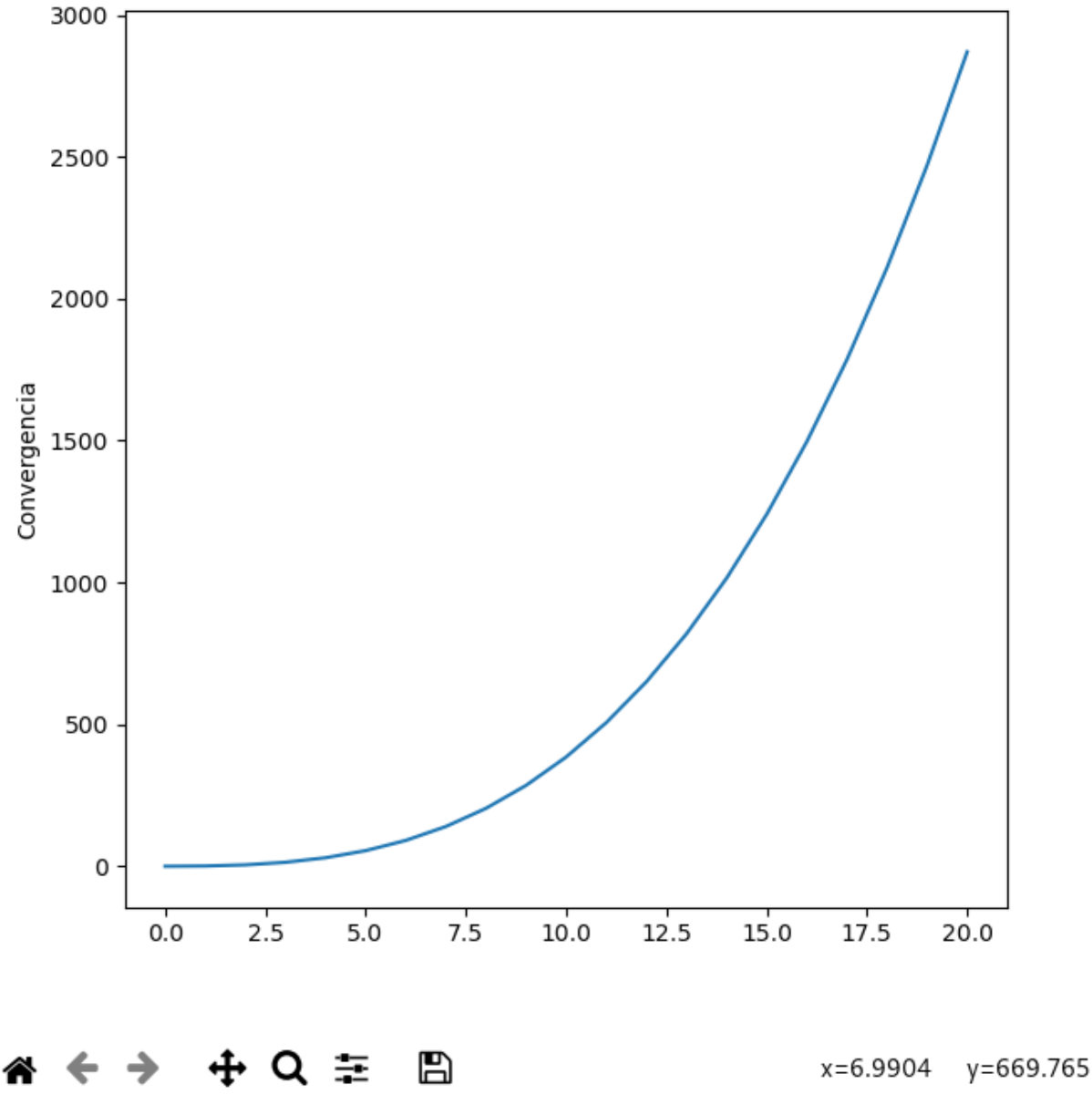
n: 12

Suma: 506



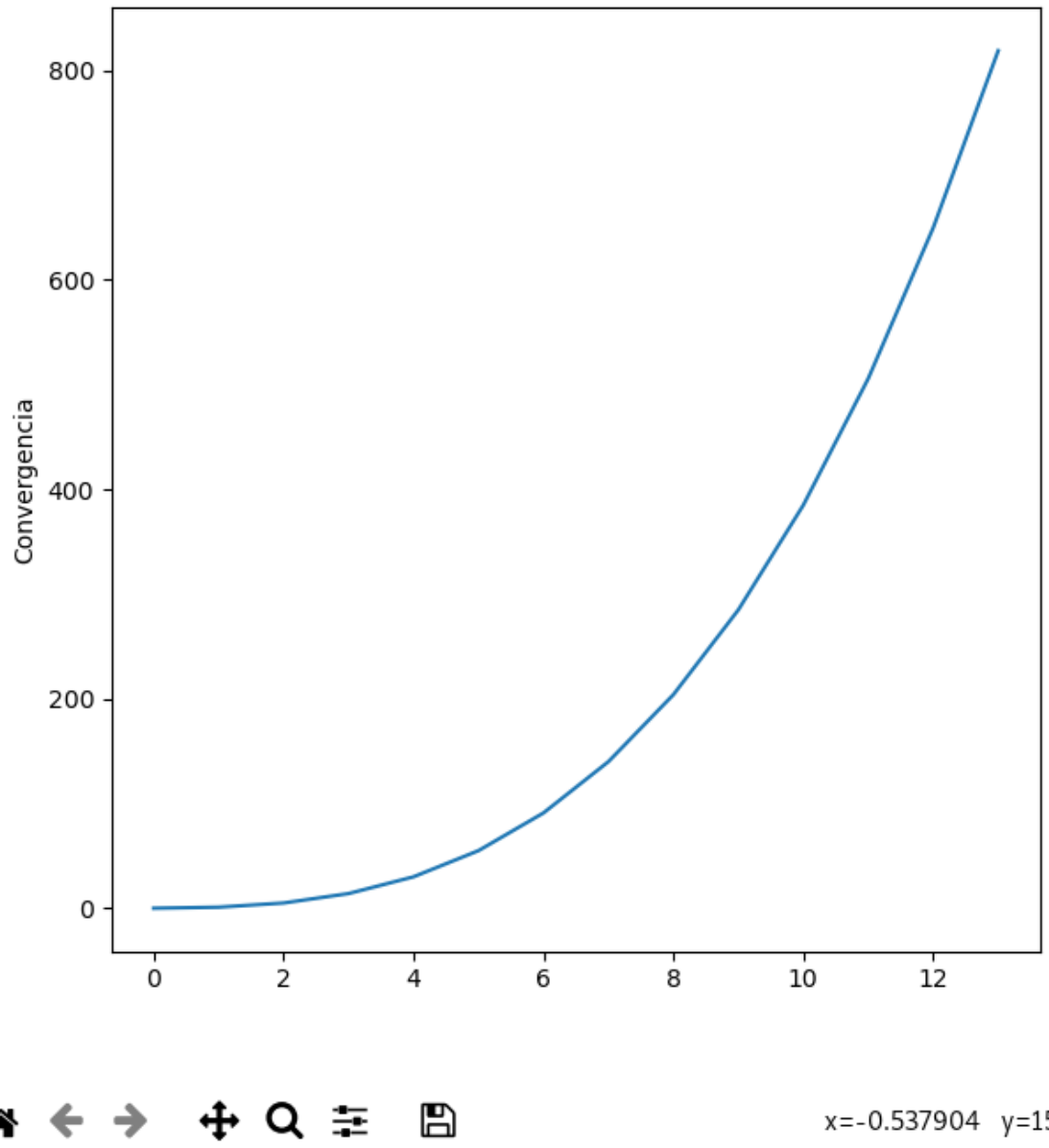
N: 21

Suma: 2870


3000
2500
2000
1500
1000
500
0
Convergencia
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
x=6.9904
y=669.765

N: 14

Suma: 819



Punto 2.

a. Para aplicar la fórmula recursiva descrita en el enunciado, se usó el siguiente código:

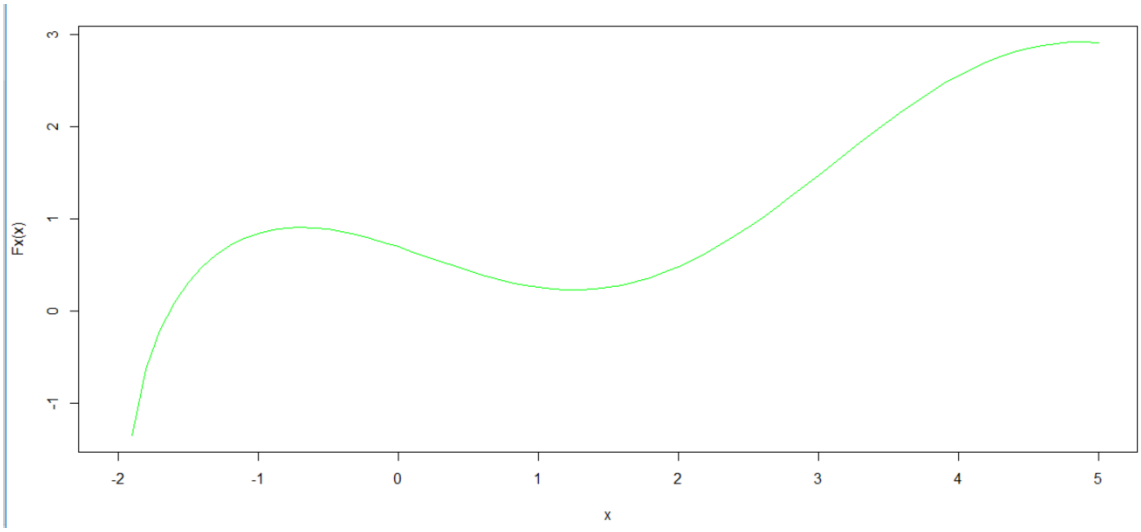
```
#Punto
2
parcial
```

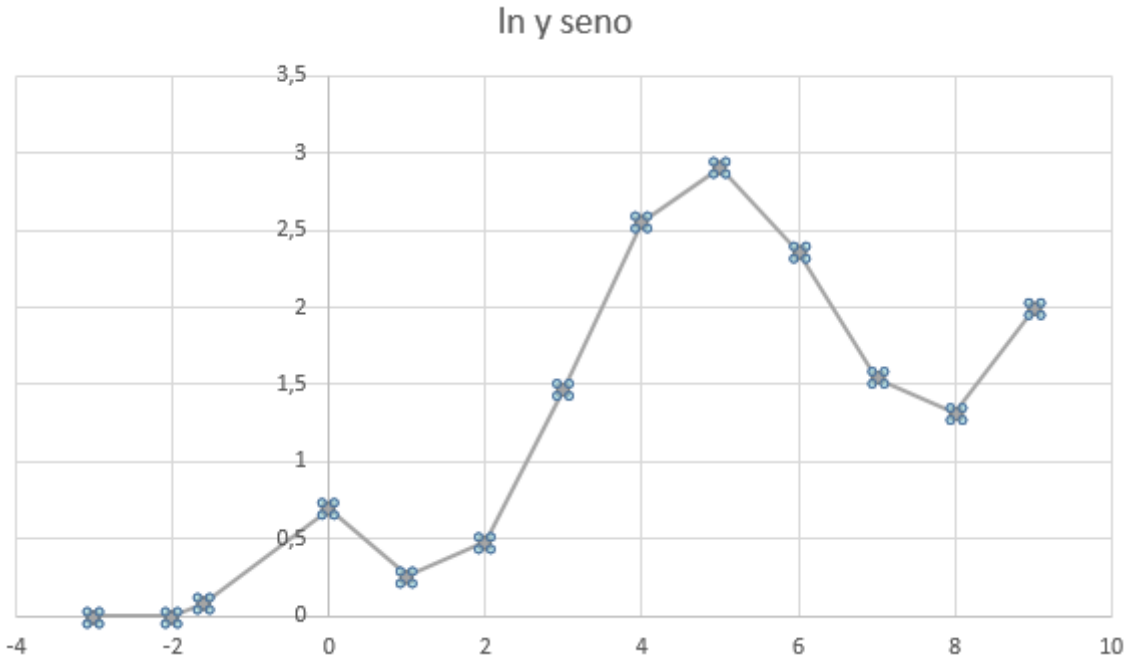
```
rm(list=ls())


Fx = function(x) log(x+2)-sin(x)


ecuacion = function(x1, x2, error){
  x = seq(-2, 5, 0.1)
  plot(x, Fx(x), type = "l", col="green")
  x = x1-((Fx(x1))*(x1-x2)) / (Fx(x1) - Fx(x2))
  err = 1
  contador = 0
  while (err > error){
    contador = contador + 1
    x1 = x2
    x2 = x
    x = x1-((Fx(x1))*(x1-x2)) / (Fx(x1) - Fx(x2))
    err = abs((x-x2)/x)*100
    cat("Valor X: ", x, "\t\tValor del Error: ", err, "\t\tIteracion: ",
contador, "\n")
  }
}


ecuacion(0,5, 10e-8)
```

En donde se obtuvo la siguiente tabla que refleja el número de iteración, el valor del X en cada una de estas, y su valor de error.

```
Valor X:  -1.957293           Valor del Error:  19.93993            Iteracion:  1
Valor X:  -1.593619           Valor del Error:  22.82065            Iteracion:  2
Valor X:  -1.609138           Valor del Error:  0.9644408           Iteracion:  3
Valor X:  -1.632711           Valor del Error:  1.443803            Iteracion:  4
Valor X:  -1.631401           Valor del Error:  0.08029122          Iteracion:  5
Valor X:  -1.631444           Valor del Error:  0.00260241          Iteracion:  6
Valor X:  -1.631444           Valor del Error:  4.980356e-06        Iteracion:  7
Valor X:  -1.631444           Valor del Error:  3.192299e-10        Iteracion:  8
```

Así mismo, se obtuvo una gráfica en donde se puede evidenciar lo dicho en la tabla anterior, la cual fue comparada con una realizada en Excel con el fin de evidenciar qué tanto error presentaban entre ambas.





b. Para aplicar el algoritmo descrito en el enunciado, se usó el siguiente código:

```
#Punto
2b
parcial
```

```
rm(list=ls())


Fx = function(x) log(x+2)-sin(x)#exp(x) - pi * x
ecuacion = function(x, x1, error){
  x = seq(-2, 5, 0.1)
    plot(x, Fx(x), type = "l", col="blue")
  x2= x1-((Fx(x1))*(x1-x)) / (Fx(x1) - Fx(x))
  err = 1
  contador = 0
  while (err > error){
    contador = contador + 1
    if(Fx(x2)&& Fx(x1)<0){
    x2= x1
    x1=x
    }
    else{
      x2=x1
    }
    x2= x1-((Fx(x1))*(x1-x)) / (Fx(x1) - Fx(x))
    err = abs((x2-x1)/x2)*100
    cat("Valor X: ", x, "\t\tValor del Error: ", err, "\t\tIteracion: ",
contador, "\n")
  }
}


ecuacion(-1,5, 10e-8)
```

En donde se muestra en la siguiente imagen, el rango de números que puede tomar X hasta 4

```
Valor X:  -2 -1.9 -1.8 -1.7 -1.6 -1.5 -1.4 -1.3 -1.2 -1.1 -1 -0.9 -0.8 -0.7 -0.6 -0.5
-0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7
1.8 1.9 2 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4
```