



Universidad Autónoma del Estado de México
Facultad de Ciencias
Licenciatura en Física
UA: Física Computacional
Periodo Escolar 2022B
Estudiantes:

- Francisco Javier De La Cruz Lugo,
- Isaac Levy Gutiérrez Monroy,
- José Abraham Rodríguez Pineda.

Catedrático: Dr. Jorge López Lemus
Título: Proyecto "Ajuste de Curvas"



Descripción:

El presente proyecto trata sobre el desarrollo de un código de programa escrito en el lenguaje de programación *Python*, el cual, tiene como objetivo realizar un ajuste de curvas lineales y no lineales para series de datos que siguen un comportamiento relacionado a un cierto tipo de función o curva. En este caso, el mismo programa se encarga de generar dichos datos de forma aleatoria, sin embargo, el uso de este tipo de aplicaciones ofrece una alternativa al momento de realizar análisis para datos obtenidos de forma experimental en un laboratorio.

Código:

```
from scipy.optimize import curve_fit
import numpy as np
from matplotlib import pyplot as plt
from random import random

#Definiendo todas funciones que vamos a utilizar

def linear (x, a ,b):
    return a*x + b

def parabola(x, a, b , c):
    return a*x*x + b * x + c

def normal(x, miu, sigma):
    return np.exp(-((x-miu)**2)*(2*sigma**2))

def exponencial (x, lamb):
    return lamb*np.exp(-lamb*x)

def weibull (x, alfa, beta, indice):
    return x**(indice)*np.exp(-(x/beta)**alfa)

def lognormal (x, miu, sigma):
    return (1/x)*np.exp(-((np.log(x)-miu)**2)/(2*sigma**2))

## Se hacen todos los procesos
opcion = 8
```

```

while (opcion != 0):
    print("""¿Que tipo de funcion quieres utilizar?
    Lineal ..... [1]
    Parabolica ..... [2]
    Normal ..... [3]
    Exponencial ..... [4]
    Weibull ..... [5]
    Lognormal ..... [6]
    Salir ..... [0]
    """)

    op = int(input("Oprime la opcion que quieres: "))

    if op == 1:
        print("Escogiste un 'Ajuste lineal':  $y = a*x + b$ ")
        print("Generando datos aleatorios")
        print("Empezamos con los intervalos (en enteros)")
        inf = int(input("x0: "))
        sup = int(input("x1: "))
        xx = np.linspace(inf, sup, 30)
        ca = float(input("Dame el coeficiente lineal a: ")) # Yo
le mando antes los valores con los
        cb = float(input("Dame el coeficiente independiente b: ")) #
que obtendr ; datos aleatorios que se acomoden
        yy = []
        for val in xx:
            evaluacion_aleatoria = linear(val, ca, cb) + (random()-0.5)*5
            yy.append(evaluacion_aleatoria)
        coeficientes, covarianza = curve_fit(linear, xx, yy)
        print("Coeficientes [a,b]:",coeficientes)
        plt.scatter(xx, yy, label = "Datos aleatorios")
        plt.plot(xx, linear(xx, coeficientes[0], coeficientes[1]), 'r--',
label = "Ajuste:  $y =$ " + str(coeficientes[0]) + " $*x +$ " +
str(coeficientes[1]))
        plt.legend()
        plt.grid()
        plt.show()

    if op == 2:
        print("Escogiste un 'Ajuste Parabolico':  $y = a*x^2 + b*x + c$ ")
        print("Generando datos aleatorios")
        print("Empezamos con los intervalos (en enteros)")
        inf = int(input("x0: "))
        sup = int(input("x2: "))
        xx = np.linspace(inf, sup, 30)
        ca = float(input("Dame el coeficiente cuadrado a: ")) #Yo le
mando antes los valores con los
        cb = float(input("Dame el coeficiente lineal b: ")) #que
obtendr ; datos aleatorios que se acomoden
        cc = float(input("Dame el coeficiente independiente c: "))
        yy = []
        for val in xx: #Para hacer datos cercanos a la parabola
            evaluacion_aleatoria = parabola(val, ca, cb, cc) + (random()-
0.5)*5

```

```

        yy.append(evaluacion_aleatoria)
        coeficientes, covarianza = curve_fit(parabola, xx, yy)
        print("Coeficientes [a,b,c]: ",coeficientes)
        plt.scatter(xx, yy, label = "Datos Aleatorios")
        plt.plot(xx, parabola(xx, coeficientes[0], coeficientes[1],
coeficientes[2]), "r--", label = "Ajuste: y = " + str(coeficientes[0]) +
"*x^2 + " + str(coeficientes[1]) + "*x + " + str(coeficientes[2]))
        plt.legend()
        plt.grid()
        plt.show()

    if op == 3:
        print("Escogiste un 'Ajuste Normal': y = exp(-(x-
miu)**2)*(2*sigma**2))")
        print("Generando datos aleatorios")
        print("Empecemos con los intervalor (En enteros)")
        inf = int(input("x0: "))
        sup = int(input("x1: "))
        numero_datos = int(input("Dame el numero de datos (Recomendamos
un valor mayor a 100): "))
        xx = np.linspace(inf, sup, numero_datos)
        miu = float(input("Dame el valor de miu: "))
        sigma = float(input("Dame el valor de sigma (El valor de sigma
debe ser positivo): "))
        yy = []
        for val in xx:
            evaluacion_aleatoria = normal(val, miu, sigma) + (random()-
0.5)*0.1
            yy.append(evaluacion_aleatoria)
            coeficientes, covarianza = curve_fit(normal, xx, yy)
            print("Coeficientes [miu,sigma]: ",coeficientes)
            plt.scatter(xx, yy, label = "Datos Aleatorios")
            plt.plot(xx, normal(xx, coeficientes[0], coeficientes[1]), "r--",
label = "Ajuste y = exp(-(x-
"+str(coeficientes[0])+")**2)*(2*"+str(coeficientes[1])+"**2))")
            plt.legend()
            plt.grid()
            plt.show()

    if op == 4:
        print("Escogiste un 'Ajuste Exponencial': y = lamb*exp(-lamb*x)")
        print("Generando Datos Aleatorios")
        print("Empecemos con los intervalo (en enteros desde 0 en
adelante)")
        inf = int(input("x0: "))
        sup = int(input("x2: "))
        numero_datos = int(input("Dame el numero de datos (Recomendamos
un valor mayor a 100): "))
        xx = np.linspace(inf, sup, numero_datos)
        lamb = float(input("Dame el valor de lambda (debe ser positivo):
"))
        yy = []
        for val in xx:

```

```

        evaluacion_aleatoria = exponencial(val, lamb) + (random()-
0.5)*0.1
        yy.append(evaluacion_aleatoria)
        coeficientes, covarianza = curve_fit(exponencial, xx, yy)
        print("Coeficiente [lambda]: ",coeficientes)
        plt.scatter(xx, yy, label = "Datos Aleatorios")
        plt.plot(xx, exponencial(xx, coeficientes[0]), "r--", label =
"Ajuste y = "+str(coeficientes[0])+"*exp(-"+str(coeficientes[0])+"*x)")
        plt.legend()
        plt.grid()
        plt.show()

    if op == 5:
        print("Escogiste un 'Ajuste de Weibull':  $y = x^{(indice)} * \exp(-(x/beta)^{alfa})$ ")
        print("Generando Datos Aleatorios")
        print("Empecemos con los intervalo (en enteros desde 0 en
adelante) ")
        inf = int(input("x0: "))
        sup = int(input("x1: "))
        numero_datos = int(input("Dame el numero de datos (Recomendamos
un valor mayor a 100): "))
        xx = np.linspace(inf, sup, numero_datos)
        alfa = float(input("Dame el valor de alfa (debe ser positivo):
"))
        beta = float(input("Dame el valor de beta (debe ser positivo):
"))
        indicealfa_menos1 = float(alfa - 1)
        yy = []
        for val in xx:
            evaluacion_aleatoria = weibull(val, alfa, beta,
indicealfa_menos1) + (random()-0.5)*0.1
            yy.append(evaluacion_aleatoria)
            coeficientes, covarianza = curve_fit(weibull, xx, yy)
            print("Coeficientes [alfa,beta,indice]: ",coeficientes)
            plt.scatter(xx, yy, label = "Datos Aleatorios")
            plt.plot(xx, weibull(xx, coeficientes[0], coeficientes[1],
coeficientes[2]), "r--", label = "Ajuste: y =
x**("+str(coeficientes[2])+")*exp(-
(x/"+str(coeficientes[1])+")**"+str(coeficientes[0])+"")")
            plt.legend()
            plt.grid()
            plt.show()

    if op == 6:
        print("Escogiste un 'Ajuste Lognormal':  $y = (1/x) * \exp(-((\log(x) - \mu)^2) / (2 * \sigma^2))$ ")
        print("Generando Datos Aleatorios")
        print("Empecemos con los intervalo (en enteros desde 0 en
adelante) ")
        inf = int(input("x0: "))
        sup = int(input("x1: "))
        numero_datos = int(input("Dame el numero de datos (Recomendamos
un valor mayor a 100): "))

```

```

xx = np.linspace(inf, sup, numero_datos)
if inf == 0:
    xx = xx[1:]
miu = float(input("Dame el valor de miu: "))
sigma = float(input("Dame el valor de sigma (debe ser positivo):
"))

yy = []
for val in xx:
    evaluacion_aleatoria = lognormal(val, miu, sigma) +
(random()-0.5)*.1
    yy.append(evaluacion_aleatoria)
    coeficientes, covarianza = curve_fit(lognormal, xx, yy)
    print("Coeficientes [miu,sigma]: ",coeficientes)
    plt.scatter(xx,yy, label = "Datos Aleatorios")
    plt.plot(xx, lognormal(xx, coeficientes[0],coeficientes[1]), "r--
", label = "Ajuste: y = (1/x)*exp(-(log(x)-
"+str(coeficientes[0])+")**2)/(2*"+str(coeficientes[1])+"**2))")
    plt.legend()
    plt.grid()
    plt.show()

opcion = op

```

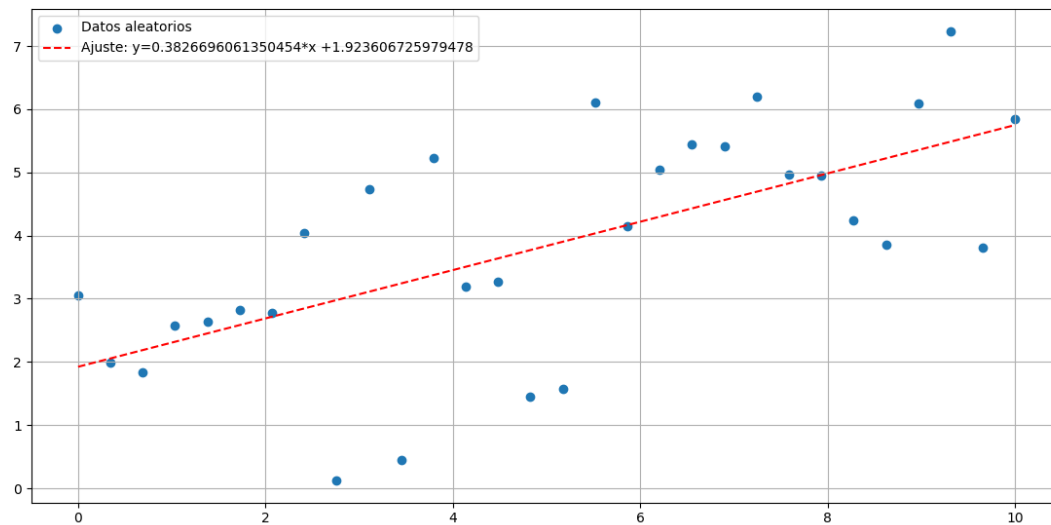
Funcionamiento:

1. Ajuste Lineal

$$y = ax + b$$

```
pakinyxd@F-Latitude-E7450:/media/pakinyxd/USB_DISK/ELPacohontas/FC/Proyecto$ python3 proyecto_ajustes.py
¿Que tipo de funcion quieres utilizar?
Lineal ..... [1]
Parabolica ..... [2]
Normal ..... [3]
Exponencial ..... [4]
Weibull ..... [5]
Lognormal ..... [6]
Salir ..... [0]

Oprime la opcion que quieres: 1
Escogiste un 'Ajuste lineal': y = a*x + b
Generando datos aleatorios
Empezamos con los intervalos (en enteros)
x0: 0
x1: 10
Dame el coeficiente lineal a: 0.5
Dame el coeficiente independiente b: 1
Coeficientes [a,b]: [0.38266961 1.92360673]
```

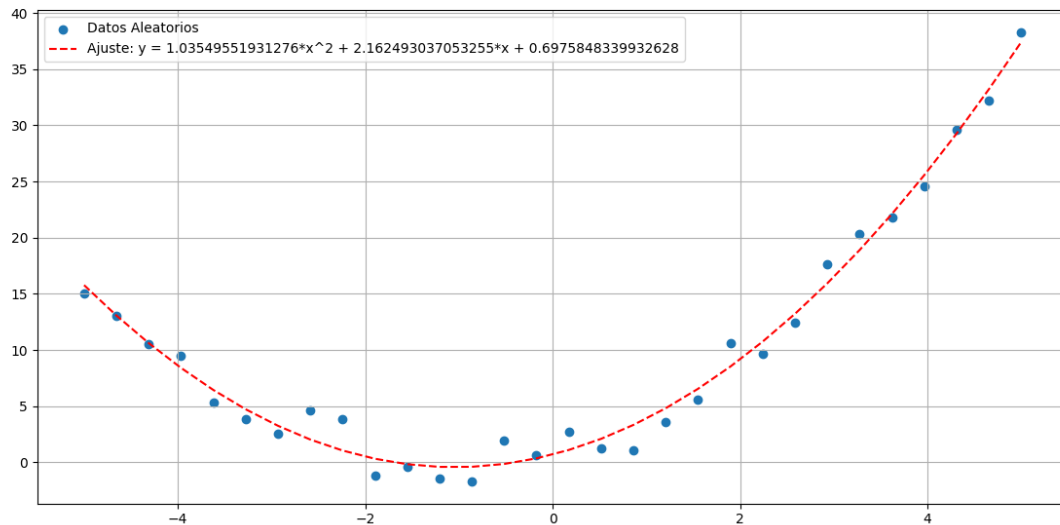


2. Ajuste Parabólico:

$$y = ax^2 + bx + c$$

```
pakinyxd@F-Latitude-E7450:/media/pakinyxd/USB_DISK/ELPacohontas/FC/Proyecto$ python3 proyecto_ajustes.py
¿Que tipo de funcion quieres utilizar?
Lineal ..... [1]
Parabolica ..... [2]
Normal ..... [3]
Exponencial ..... [4]
Weibull ..... [5]
Lognormal ..... [6]
Salir ..... [0]

Oprime la opcion que quieres: 2
Escogiste un 'Ajuste Parabolico': y = a*x^2 + b*x + c
Generando datos aleatorios
Empezamos con los intervalos (en enteros)
x0: -5
x2: 5
Dame el coeficiente cuadrado a: 1
Dame el coeficiente lineal b: 2
Dame el coeficiente independiente c: 1
Coeficientes [a,b,c]: [1.03549552 2.16249304 0.69758483]
```

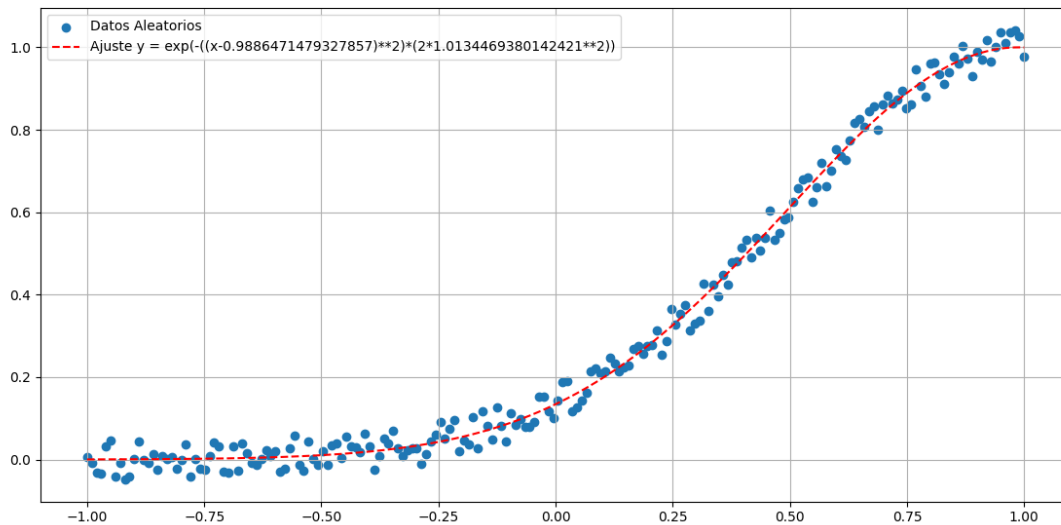


3. Ajuste Normal:

$$y = e^{-2(x-\mu)^2\sigma^2}$$

```
pakinyxd@F-Latitude-E7450:/media/pakinyxd/USB_DISK/ELPacohontas/FC/Proyecto$ python3 proyecto_ajustes.py
¿Que tipo de funcion quieres utilizar?
Lineal ..... [1]
Parabolica ..... [2]
Normal ..... [3]
Exponencial ..... [4]
Weibull ..... [5]
Lognormal ..... [6]
Salir ..... [0]

Oprime la opcion que quieres: 3
Escogiste un 'Ajuste Normal': y = exp(-((x-miu)**2)*(2*sigma**2))
Generando datos aleatorios
Empecemos con los intervalor (En enteros)
x0: -1
x1: 1
Dame el numero de datos (Recomendamos un valor mayor a 100): 200
Dame el valor de miu: 1
Dame el valor de sigma (El valor de sigma debe ser positivo): 1
Cofecientes [miu,sigma]: [0.98864715 1.01344694]
```

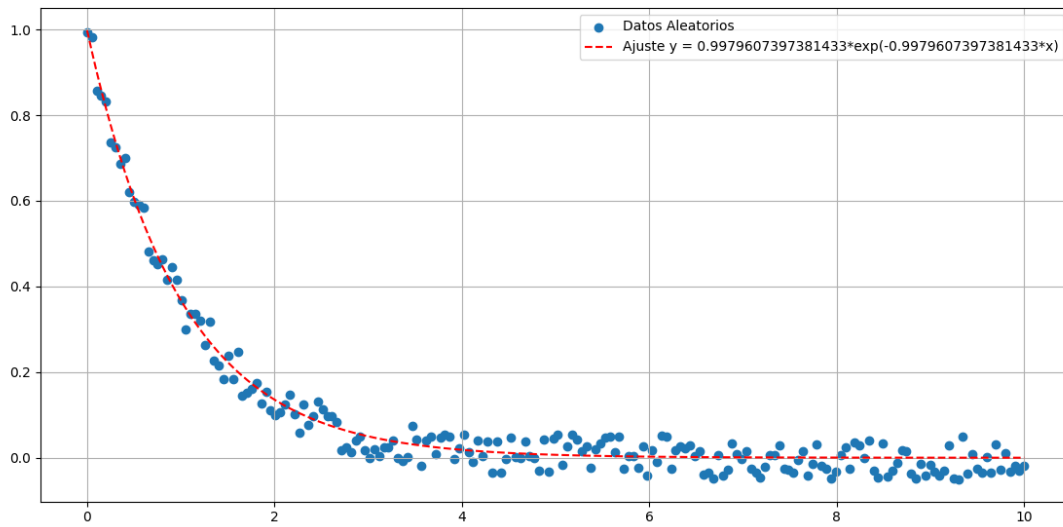


4. Ajuste Exponencial:

$$y = \lambda e^{-\lambda x}$$

```
Oprime la opcion que quieres: 0
pakinyxd@F-Latitude-E7450:/media/pakinyxd/USB DISK/ELPacohontas/FC/Proyecto$ python3 proyecto_ajustes.py
¿Que tipo de funcion quieres utilizar?
Lineal ..... [1]
Parabolica ..... [2]
Normal ..... [3]
Exponencial ..... [4]
Weibull ..... [5]
Lognormal ..... [6]
Salir ..... [0]

Oprime la opcion que quieres: 4
Escogiste un 'Ajuste Exponencial': y = lamb*exp(-lamb*x)
Generando Datos Aleatorios
Empecemos con los intervalo (en enteros desde 0 en adelante)
x0: 0
x2: 10
Dame el numero de datos (Recomendamos un valor mayor a 100): 200
Dame el valor de lambda (debe ser positivo): 1
Coeficiente [lambda]: [0.99796074]
```

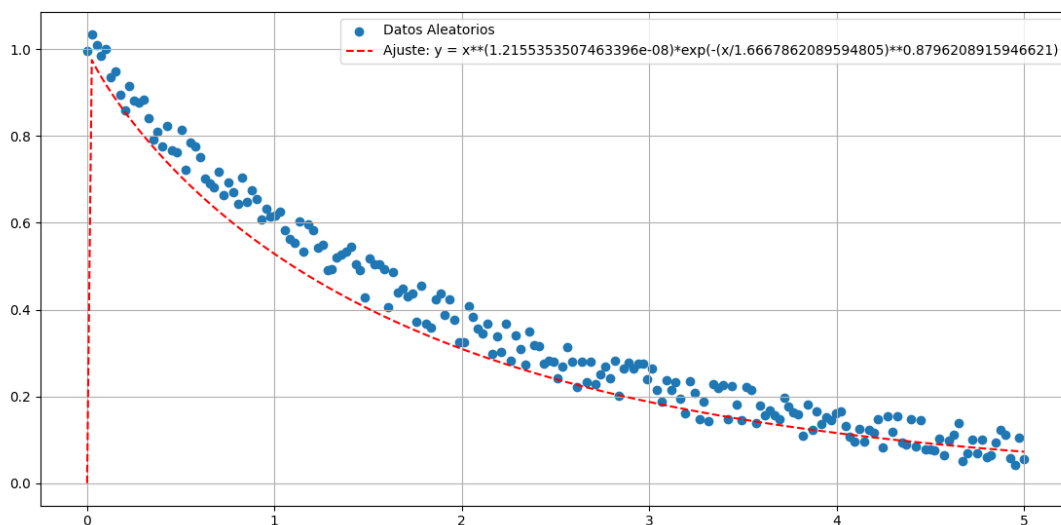


5. Ajuste Weibull:

$$y = x^i e^{-\left(\frac{x}{\beta}\right)^\alpha}$$

```
pakinyxd@F-Latitude-E7450:/media/pakinyxd/USB DISK/ELPacohontas/FC/Proyecto$ python3 proyecto_ajustes.py
¿Que tipo de funcion quieres utilizar?
Lineal ..... [1]
Parabolica ..... [2]
Normal ..... [3]
Exponencial ..... [4]
Weibull ..... [5]
Lognormal ..... [6]
Salir ..... [0]

Oprime la opcion que quieres: 5
Escogiste un 'Ajuste de Weibull': y = x**(indice)*exp(-(x/beta)**alfa)
Generando Datos Aleatorios
Empecemos con los intervalo (en enteros desde 0 en adelante)
x0: 0
x1: 5
Dame el numero de datos (Recomendamos un valor mayor a 100): 200
Dame el valor de alfa (debe ser positivo): 1
Dame el valor de beta (debe ser positivo): 2
/media/pakinyxd/USB DISK/ELPacohontas/FC/Proyecto/proyecto_ajustes.py:21: RuntimeWarning: divide by zero
  return x**(indice)*np.exp(-(x/beta)**alfa)
Coeficientes [alfa,beta,indice]: [8.79620892e-01 1.66678621e+00 1.21553535e-08]
```



6. Ajuste Lognormal:

$$y = \frac{1}{x} e^{-(\log(x)-\mu)^2/2\sigma^2}$$

```
pakinyxd@F-Latitude-E7450:/media/pakinyxd/USB_DISK/ELPacohontas/FC/Proyecto$ python3 proyecto_ajustes.py
¿Que tipo de funcion quieres utilizar?
Lineal ..... [1]
Parabolica ..... [2]
Normal ..... [3]
Exponencial ..... [4]
Weibull ..... [5]
Lognormal ..... [6]
Salir ..... [0]

Oprime la opcion que quieres: 6
Escogiste un 'Ajuste Lognormal': y = (1/x)*exp(-((log(x)-miu)**2)/(2*sigma**2))
Generando Datos Aleatorios
Empecemos con los intervalo (en enteros desde 0 en adelante)
x0: 0
x1: 10
Dame el numero de datos (Recomendamos un valor mayor a 100): 200
Dame el valor de miu: -1
Dame el valor de sigma (debe ser positivo): 1
Coeficientes [miu,sigma]: [-0.99599272  1.00119711]
¿Que tipo de funcion quieres utilizar?
```

