

Tarea 8

Boris Garcés

Tabla de Contenidos

Mínimos cuadrados	1
a)	2
b)	3
C)	4
D)	6
e)	7
a)	9
b)	10
c)	11
d)	12
e)	13

Mínimos cuadrados

1. Dados los datos:

x_i	4.0	4.2	4.5	4.7	5.1	5.5	5.9	6.3	6.8	7.1
y_i	102.56	130.11	113.18	142.05	167.53	195.14	224.87	256.73	299.50	326.72

- Construya el polinomio por mínimos cuadrados de grado 1 y calcule el error.
- Construya el polinomio por mínimos cuadrados de grado 2 y calcule el error.
- Construya el polinomio por mínimos cuadrados de grado 3 y calcule el error.
- Construya el polinomio por mínimos cuadrados de la forma be^{ax} y calcule el error.
- Construya el polinomio por mínimos cuadrados de la forma bx^a y calcule el error.

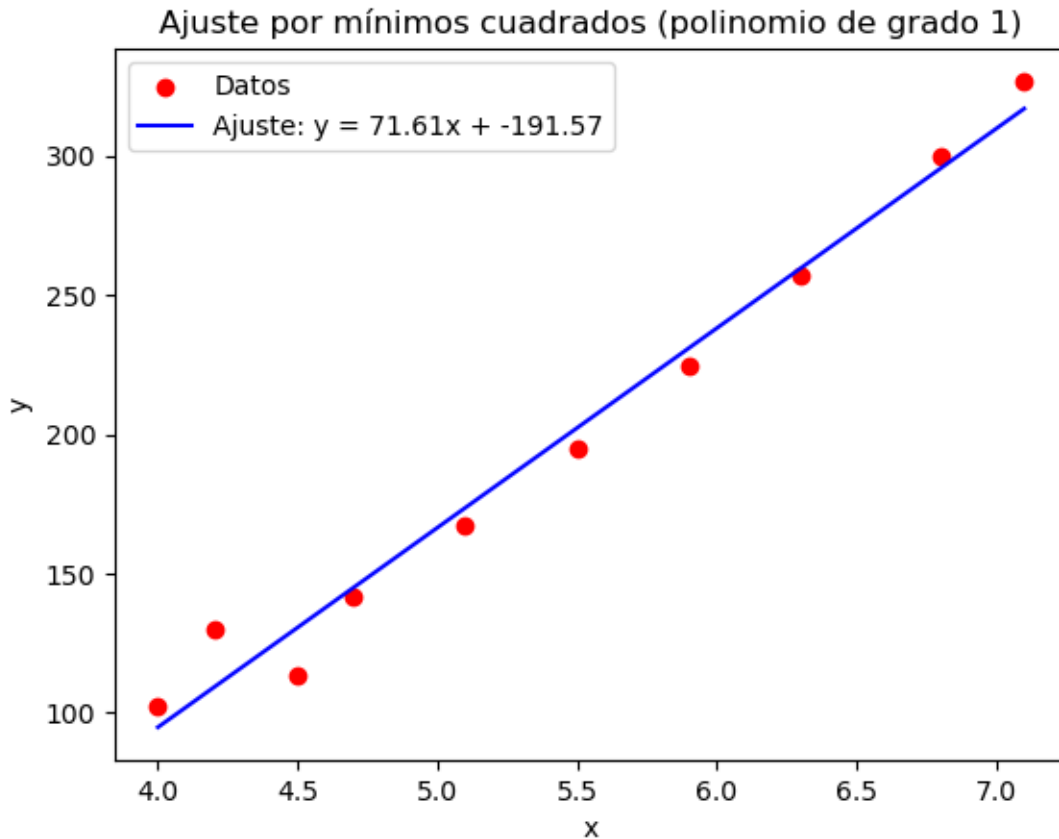
Figura 1: c1

a)

```
def error(y,ypredicho):  
    return np.sum((y-ypredicho)**2)
```

```
import numpy as np  
import matplotlib.pyplot as plt  
x = np.array([4.0,4.2,4.5,4.7,5.1,5.5,5.9,6.3,6.8,7.1])  
y = np.array([102.56,130.11,113.18,142.05,167.53,195.14,224.87,256.73,299.50,326.72])  
coeficientes = np.polyfit(x, y, 1)  
a, b = coeficientes  
print(f'Los coeficientes son m = {a:.4f}, b = {b:.4f}')  
y_ajustado = np.polyval(coeficientes, x)  
sse = error(y, y_ajustado)  
print(f'La suma de los cuadrados del error es: {sse:.4f}')  
x_fit = np.linspace(min(x), max(x), 100)  
y_fit = np.polyval(coeficientes, x_fit)  
plt.scatter(x, y, color='red', label='Datos')  
plt.plot(x_fit, y_fit, color='blue', label=f'Ajuste: y = {a:.2f}x + {b:.2f}')  
plt.xlabel('x')  
plt.ylabel('y')  
plt.title('Ajuste por mínimos cuadrados (polinomio de grado 1)')  
plt.legend()  
plt.show()
```

Los coeficientes son m = 71.6102, b = -191.5724
La suma de los cuadrados del error es: 1058.8389

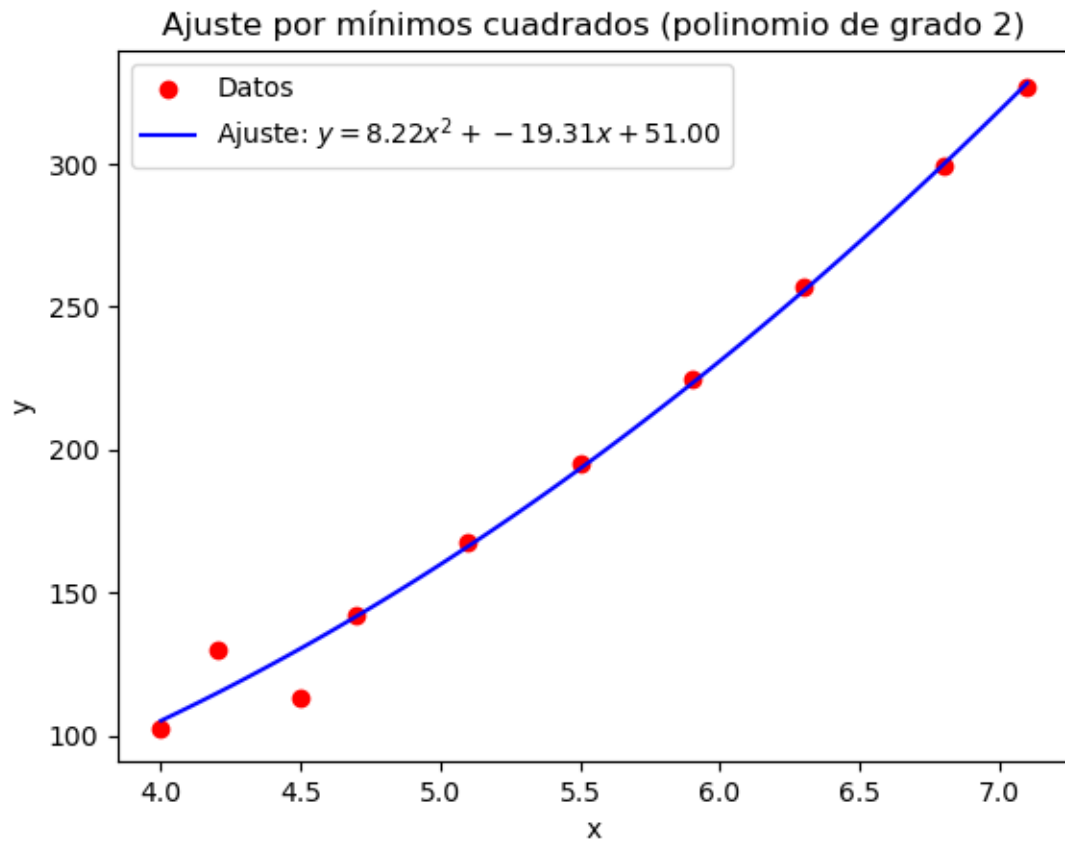


b)

```
coeficientes = np.polyfit(x, y, 2)
a, b, c = coeficientes
print(f"Coeficientes: a = {a:.4f}, b = {b:.4f}, c = {c:.4f}")
y_ajustado = np.polyval(coeficientes, x)
sse = error(y, y_ajustado)
print(f'La suma de los cuadrados del error es: {sse:.4f}')
x_fit = np.linspace(min(x), max(x), 100)
y_fit = np.polyval(coeficientes, x_fit)
plt.scatter(x, y, color='red', label='Datos')
plt.plot(x_fit, y_fit, color='blue', label=f'Ajuste: $y = {a:.2f}x^2 + {b:.2f}x + {c:.2f}$')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Ajuste por mínimos cuadrados (polinomio de grado 2)')
```

```
plt.legend()
plt.show()
```

Coeficientes: $a = 8.2171$, $b = -19.3086$, $c = 51.0008$
La suma de los cuadrados del error es: 551.6562



c)

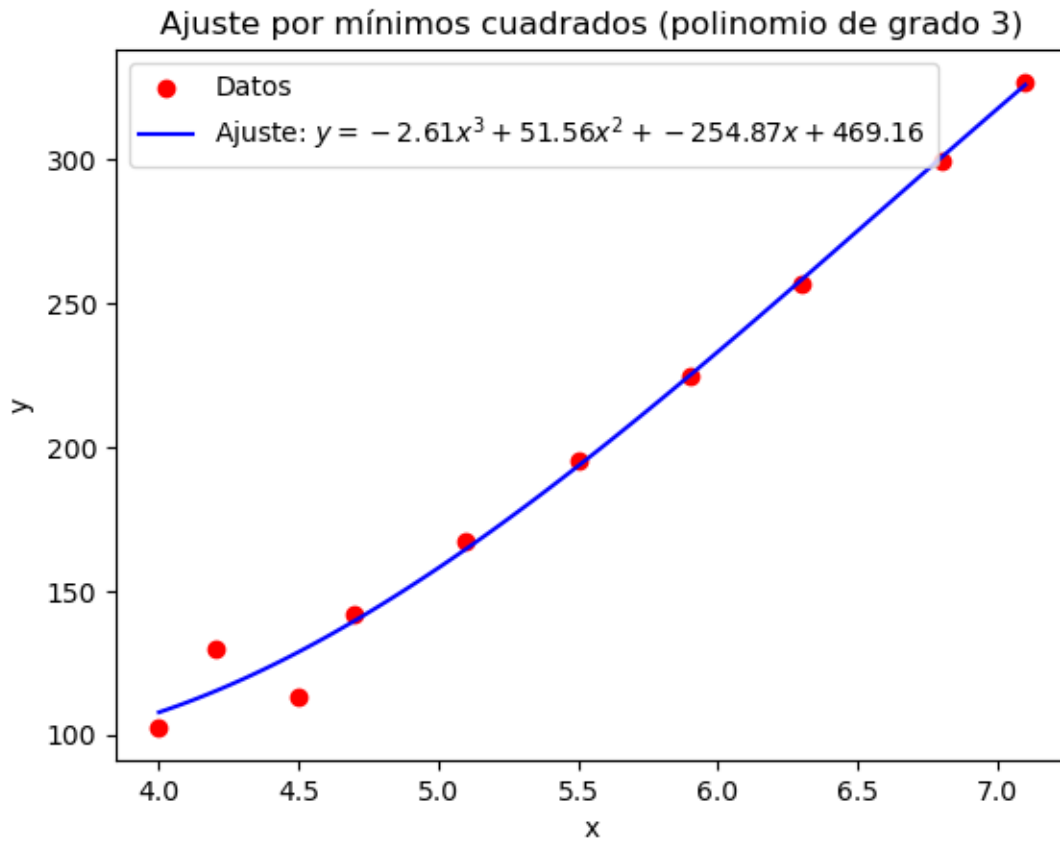
```
coeficientes = np.polyfit(x, y, 3)
a, b, c, d = coeficientes
print(f"Coeficientes: a = {a:.4f}, b = {b:.4f}, c = {c:.4f}, d = {d:.4f}")
y_ajustado = np.polyval(coeficientes, x)
sse = error(y, y_ajustado)
print(f'La suma de los cuadrados del error es: {sse:.4f}')
```

```

x_fit = np.linspace(min(x), max(x), 100)
y_fit = np.polyval(coeficientes, x_fit)
plt.scatter(x, y, color='red', label='Datos')
plt.plot(x_fit, y_fit, color='blue', label=f'Ajuste: $y = {a:.2f}x^3 + {b:.2f}x^2 + {c:.2f}x + {d:.2f}$')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Ajuste por mínimos cuadrados (polinomio de grado 3)')
plt.legend()
plt.show()

```

Coeficientes: a = -2.6068, b = 51.5610, c = -254.8748, d = 469.1633
 La suma de los cuadrados del error es: 518.3831



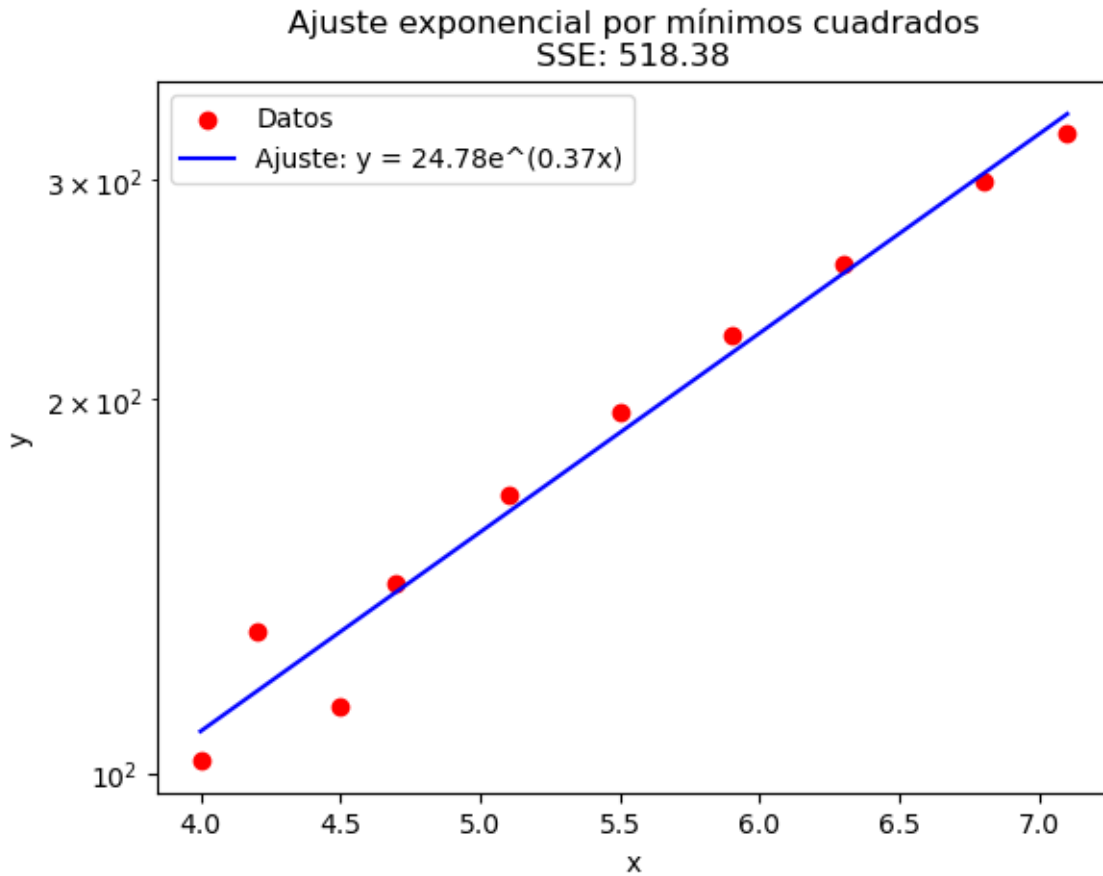
D)

Para resolver este ejercicio debemos considerar que el polinomio cumple con la forma $y = be^{ax}$ al aplicar la operación logaritmo natural a ambos lados de la ecuación podemos obtener $\ln(y) = \ln(b) + ax$

```
z=np.log(y)
coeficientes = np.polyfit(x, z, 1)
a = coeficientes[0]
ln_b= coeficientes[1]
b= np.exp(ln_b)
print(f'Los coeficientes son m = {a:.4f}, b = {b:.4f}')
sse = error(y, y_ajustado)
print(f"La suma de los cuadrados del error (SSE) es: {sse:.4f}")
x_fit = np.linspace(min(x), max(x), 100)
y_fit = b * np.exp(a * x_fit)
plt.scatter(x, y, color='red', label='Datos')
plt.plot(x_fit, y_fit, color='blue', label=f'Ajuste: y = {b:.2f}e^{a:.2f}x')
plt.xlabel('x')
plt.ylabel('y')
plt.title(f'Ajuste exponencial por mínimos cuadrados\nSSE: {sse:.2f}')
plt.legend()
plt.yscale('log')
plt.show()
```

Los coeficientes son m = 0.3685, b = 24.7767

La suma de los cuadrados del error (SSE) es: 518.3831



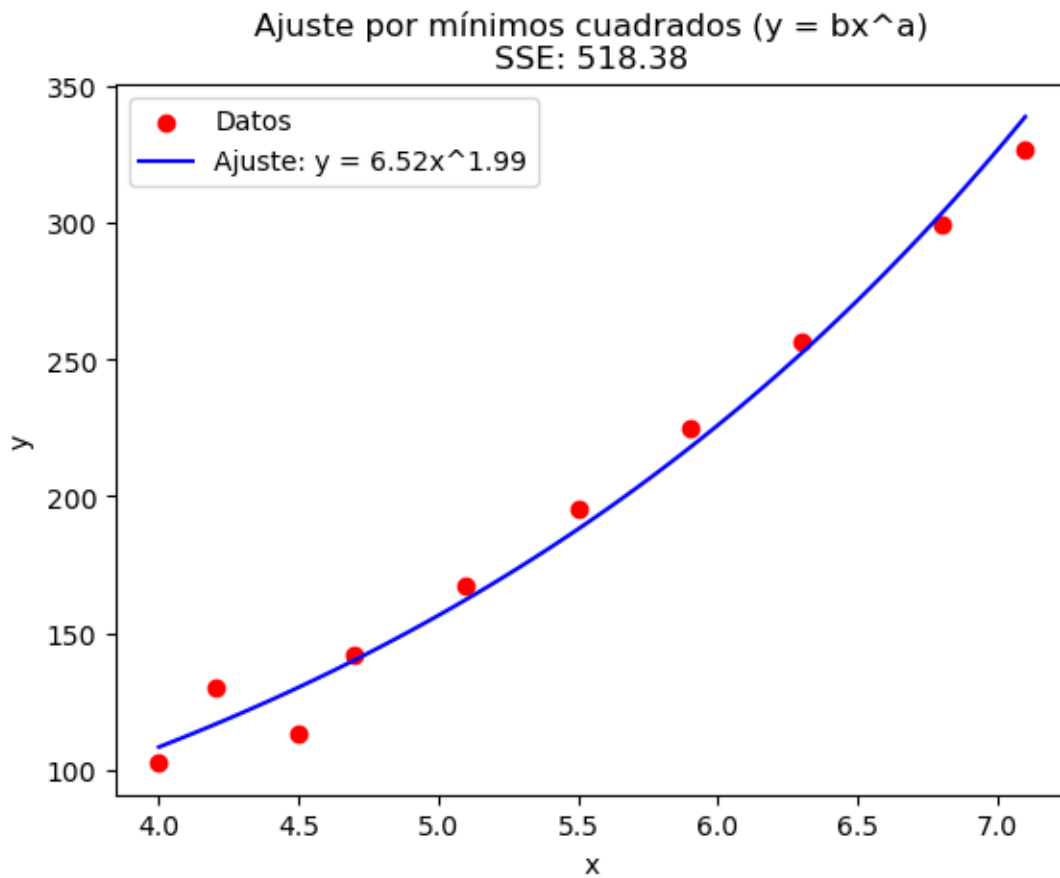
e)

Al igual que el ejercicio anterior determinaremos el logaritmo a ambos lados de la ecuación obteniendo de este modo la siguiente ecuación equivalente $\ln(y) = \ln(b) + a\ln(x)$

```
ln_x=np.log (x)
ln_y = np.log (y)
coeficientes = np.polyfit(ln_x,ln_y,1)
a = coeficientes[0]
ln_b = coeficientes[1]
b = np.exp(ln_b)
print(f'Los coeficientes son a = {a} , b = {b}')
sse = error(y, y_ajustado)
print(f"La suma de los cuadrados del error es: {sse:.4f}")
plt.scatter(x, y, color='red', label='Datos')
plt.plot(x_fit, y_fit, color='blue', label=f'Ajuste: y = {b:.2f}x^{a:.2f}')
```

```
plt.xlabel('x')
plt.ylabel('y')
plt.title(f'Ajuste por mínimos cuadrados ( $y = bx^a$ )\nSSE: {sse:.2f}')
plt.legend()
plt.show()
```

Los coeficientes son $a = 1.993284578947899$, $b = 6.518682345785465$
 La suma de los cuadrados del error es: 518.3831



2. Repita el ejercicio 5 para los siguientes datos.

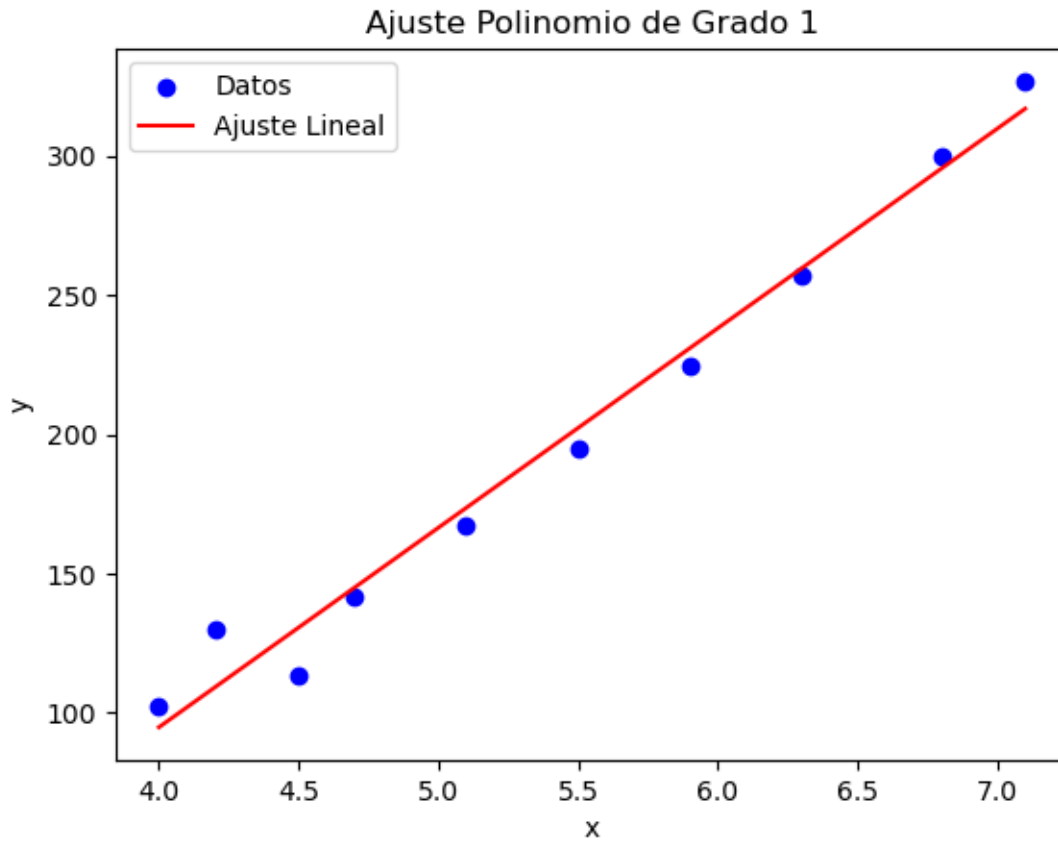
x_i	0.2	0.3	0.6	0.9	1.1	1.3	1.4	1.6
y_i	0.050446	0.098426	0.33277	0.72660	1.0972	1.5697	1.8487	2.5015


```
x2 = np.array([0.2, 0.3, 0.6, 0.9, 1.1, 1.3, 1.4, 1.6])
y2 = np.array([0.050446, 0.098426, 0.33277, 0.72660, 1.0972, 1.5697, 1.8487, 2.5015])
```

a)

```
p1 = np.polyfit(x, y, 1)
y_pred1 = np.polyval(p1, x)
error1 = error(y, y_pred1)
print(f"Polinomio de grado 1: Coeficientes = {p1}, Error = {error1}")
plt.scatter(x, y, label="Datos", color="blue")
plt.plot(x, y_pred1, label="Ajuste Lineal", color="red")
plt.xlabel("x")
plt.ylabel("y")
plt.title("Ajuste Polinomio de Grado 1")
plt.legend()
plt.show()
```

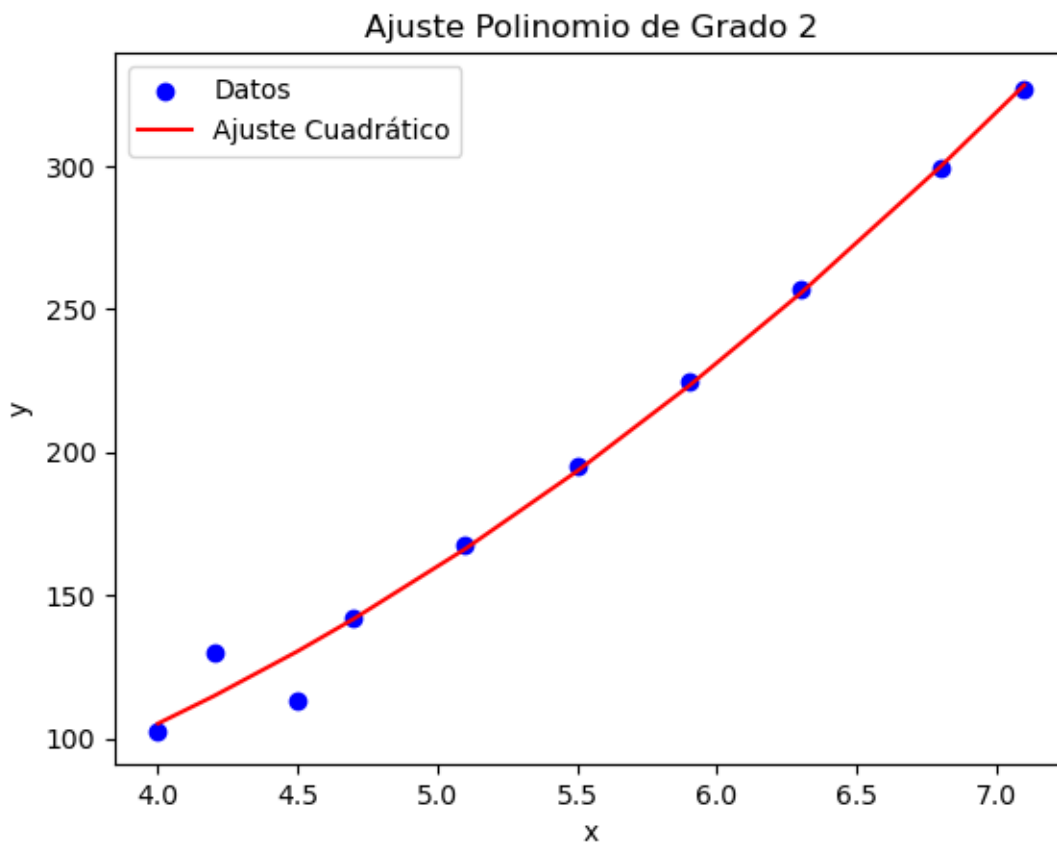
```
Polinomio de grado 1: Coeficientes = [ 71.61024372 -191.57241853], Error = 1058.83888626388
```



b)

```
p2 = np.polyfit(x, y, 2)
y_pred2 = np.polyval(p2, x)
error2 = error(y, y_pred2)
print(f"Polinomio de grado 2: Coeficientes = {p2}, Error = {error2}")
plt.scatter(x, y, label="Datos", color="blue")
plt.plot(x, y_pred2, label="Ajuste Cuadrático", color="red")
plt.xlabel("x")
plt.ylabel("y")
plt.title("Ajuste Polinomio de Grado 2")
plt.legend()
plt.show()
```

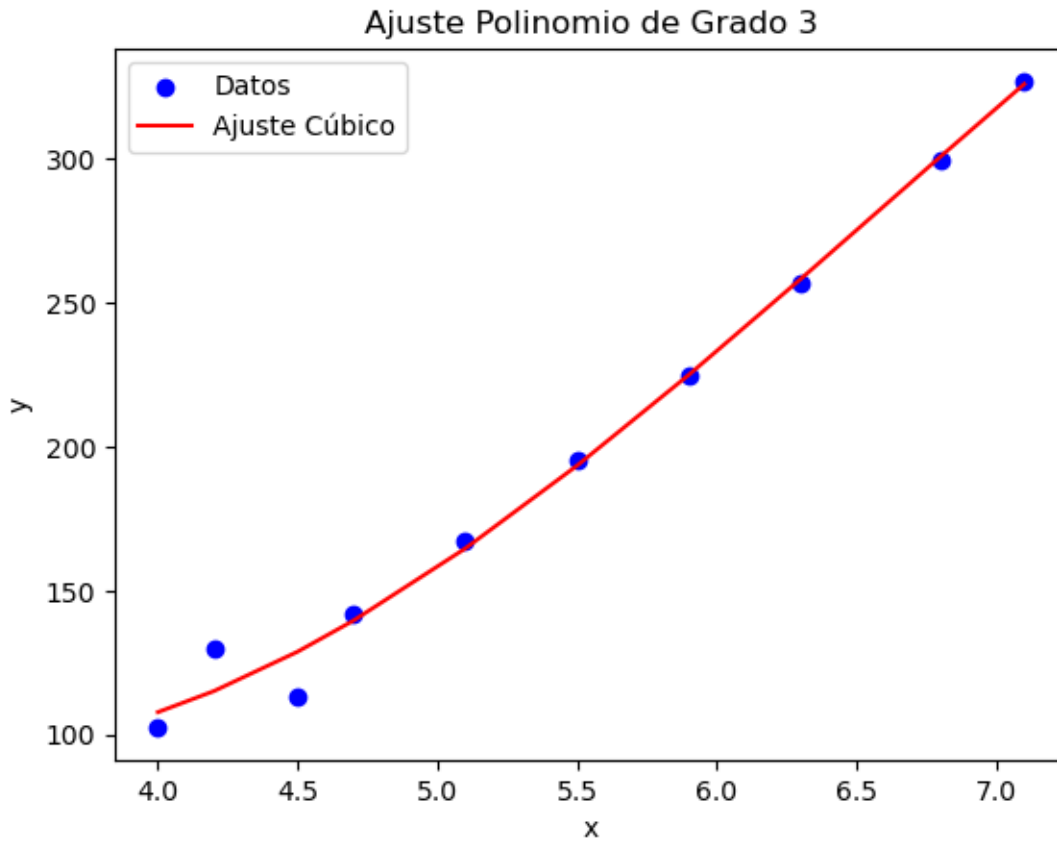
Polinomio de grado 2: Coeficientes = [8.21707232 -19.30860379 51.00078939], Error = 551.6



c)

```
p3 = np.polyfit(x, y, 3)
y_pred3 = np.polyval(p3, x)
error3 = error(y, y_pred3)
print(f"Polinomio de grado 3: Coeficientes = {p3}, Error = {error3}")
plt.scatter(x, y, label="Datos", color="blue")
plt.plot(x, y_pred3, label="Ajuste Cúbico", color="red")
plt.xlabel("x")
plt.ylabel("y")
plt.title("Ajuste Polinomio de Grado 3")
plt.legend()
plt.show()
```

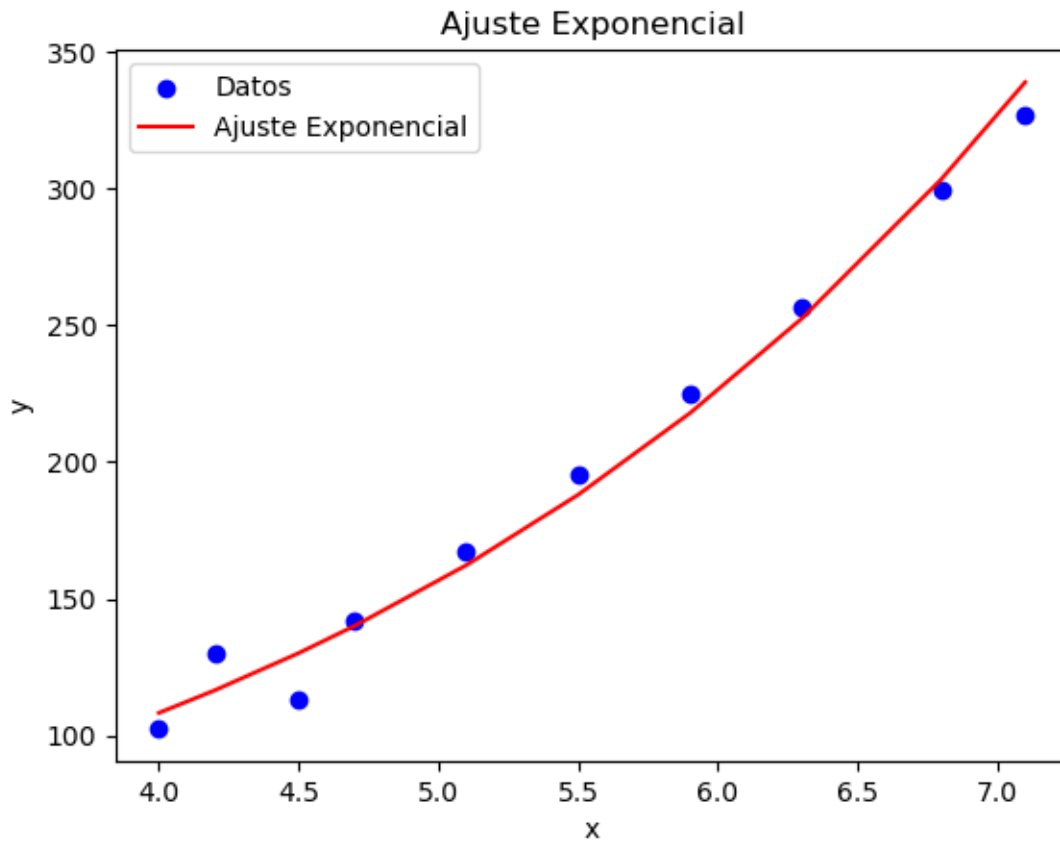
Polinomio de grado 3: Coeficientes = [-2.60683872 51.56095694 -254.87478338 469.1632652]



d)

```
y_log = np.log(y)
p_exp = np.polyfit(x, y_log, 1)
a = p_exp[0]
b = np.exp(p_exp[1])
y_pred_exp = b * np.exp(a * x)
error_exp = error(y, y_pred_exp)
print(f"Ajuste Exponencial: a = {a}, b = {b}, Error = {error_exp}")
plt.scatter(x, y, label="Datos", color="blue")
plt.plot(x, y_pred_exp, label="Ajuste Exponencial", color="red")
plt.xlabel("x")
plt.ylabel("y")
plt.title("Ajuste Exponencial")
plt.legend()
plt.show()
```

Ajuste Exponencial: $a = 0.36847662383170704$, $b = 24.776723697836218$, Error = 821.00510925779

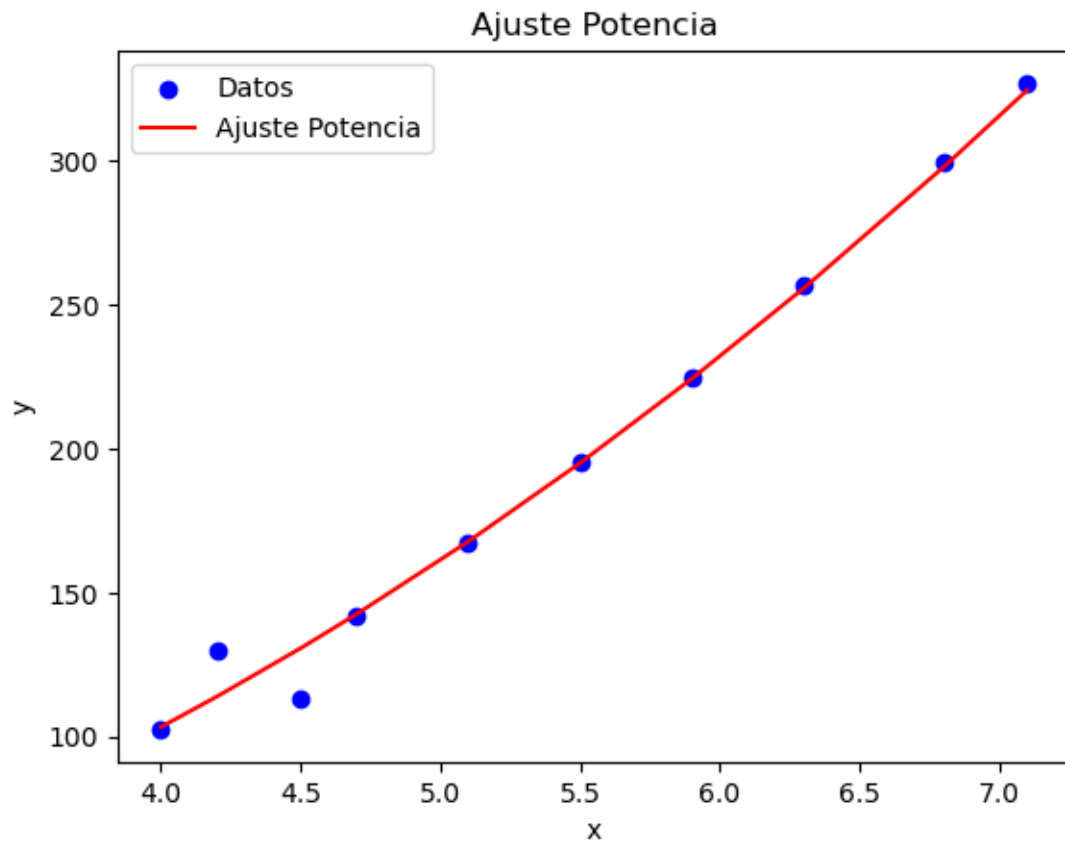


e)

```
x_log = np.log(x)
y_log = np.log(y)
p_pow = np.polyfit(x_log, y_log, 1)
a = p_pow[0]
b = np.exp(p_pow[1])
y_pred_pow = b * x**a
error_pow = error(y, y_pred_pow)
print(f"Ajuste Potencia: a = {a}, b = {b}, Error = {error_pow}")
plt.scatter(x, y, label="Datos", color="blue")
plt.plot(x, y_pred_pow, label="Ajuste Potencia", color="red")
plt.xlabel("x")
```

```
plt.ylabel("y")
plt.title("Ajuste Potencia")
plt.legend()
plt.show()
```

Ajuste Potencia: $a = 1.993284578947899$, $b = 6.518682345785465$, Error = 581.557272601367



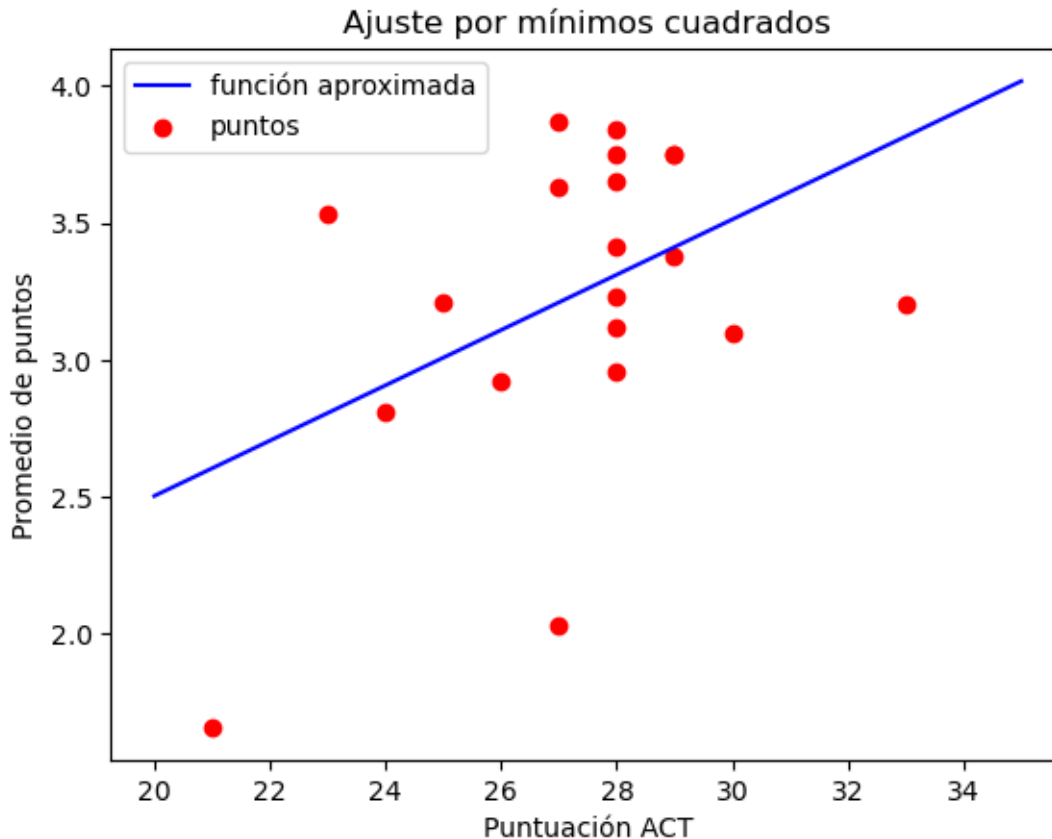
La siguiente tabla muestra los promedios de puntos del colegio de 20 especialistas en matemáticas y ciencias computacionales, junto con las calificaciones que recibieron estos estudiantes en la parte de matemáticas de la prueba ACT (Programa de Pruebas de Colegios Americanos) mientras estaban en secundaria. Grafique estos datos y encuentre la ecuación de la recta por mínimos cuadrados para estos datos.

Puntuación ACT	Promedio de puntos	Puntuación ACT	Promedio de puntos
28	3.84	29	3.75
25	3.21	28	3.65
28	3.23	27	3.87
27	3.63	29	3.75
28	3.75	21	1.66
33	3.20	28	3.12
28	3.41	28	2.96
29	3.38	26	2.92
23	3.53	30	3.10
27	2.03	24	2.81

Figura 2: c3

```
x1=np.array([28, 25, 28, 27, 28, 33, 28, 29, 23, 27,
              29, 28, 27, 29, 21, 28, 28, 26, 30, 24])
y1=np.array([3.84,3.21,3.23,3.63,3.75,3.20,3.41,3.38,3.53,2.03,
              3.75,3.65,3.87,3.75,1.66,3.12,2.96,2.92,3.10,2.81])
coeficientes= np.polyfit(x1,y1,1)
x_fit = np.linspace(20, 35, 100)
y_fit = np.polyval(coeficientes, x_fit)
a, b = coeficientes
print(f'Los coeficientes son m = {a:.4f}, b = {b:.4f}')
plt.plot(x_fit, y_fit,label='función aproximada',color='blue')
plt.scatter(x1,y1,label='puntos',color='red')
plt.xlabel('Puntuación ACT')
plt.ylabel('Promedio de puntos')
plt.title('Ajuste por mínimos cuadrados')
plt.legend(loc='upper left')
plt.show()
```

Los coeficientes son m = 0.1009, b = 0.4866



El siguiente conjunto de datos, presentado al Subcomité Antimonopolio del Senado, muestra las características comparativas de supervivencia durante un choque de automóviles de diferentes clases. Encuentre la recta por mínimos cuadrados que aproxima estos datos (la tabla muestra el porcentaje de vehículos que participaron en un accidente en los que la lesión más grave fue fatal o seria).

Tipo	Peso promedio	Porcentaje de presentación
1. Regular lujoso doméstico	4800 lb	3.1
2. Regular intermediario doméstico	3700 lb	4.0
3. Regular económico doméstico	3400 lb	5.2
4. Compacto doméstico	2800 lb	6.4
5. Compacto extranjero	1900 lb	9.6

```

peso = np.array([4800, 3700, 3400, 2800, 1900])
porcentaje= np.array([3.1, 4.0, 5.2, 6.4, 9.6])
coeficientes= np.polyfit(peso,porcentaje,1)
print(f"    Ecuación: y = {coeficientes[0]:.4f}x + {coeficientes[1]:.4f}")
x_valores=np.linspace(1900,4800,100)
y_valores = np.polyval(coeficientes, x_valores)

```



```
plt.plot(x_valores,y_valores,label="Mínimos cuadrados grado 1", color="blue")
plt.scatter(peso,porcentaje,label="puntos",color="red")
plt.xlabel("Peso")
plt.ylabel("Porcentaje")
plt.title("Ajuste a mínimos cuadrados")
plt.legend(loc="upper left")
```

Ecuación: $y = -0.0023x + 13.1465$

