| Nombre de la práctica | Regresión Lineal | | | No. | 4 |
|---|---|---|---|---|---|
| Asignatura: | Simulación | Carrera: | Ingeniería en Sistemas Computacionales | Duración de la práctica (Hrs) | |

**NOMBRE DEL ALUMNO: Vanesa Hernández Martínez**
**GRUPO:** 3501

**II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):**
Actividades en aula de clases y en equipo personal

**III. Material empleado:**
- Laptop
- Anaconda

# Regresión Lineal

La **regresión lineal** es un método estadístico que busca encontrar la relación entre dos variables, una independiente (predictora) y otra dependiente (lo que se quiere predecir). El objetivo es trazar una **línea recta** que se ajuste lo mejor posible a los datos, con la ecuación:

$$y=mx+b$$

- **y**: Valor predicho.
- **x**: Variable independiente.
- **m**: Pendiente (cambio en "y" por cada cambio en "x").
- **b**: Intersección con el eje Y (valor de "y" cuando "x" es 0).

La regresión lineal ayuda a hacer predicciones basadas en datos y se utiliza en casos como estimar precios o predecir tendencias.

# Regresión Lineal: Costo de un incidente de seguridad.

En este ejercicio se explican los fundamentos básicos de la regresión lineal aplicada a un caso de uso sencillo, relacionado con la ciberseguridad.

## Enunciado del ejercicio

El ejercicio consiste en predecir el costo de un incidente de seguridad en base al número de equipos que se han visto afectados. El conjunto de datos es generado de manera aleatoria.

## 1.- Generacion del DataSet.

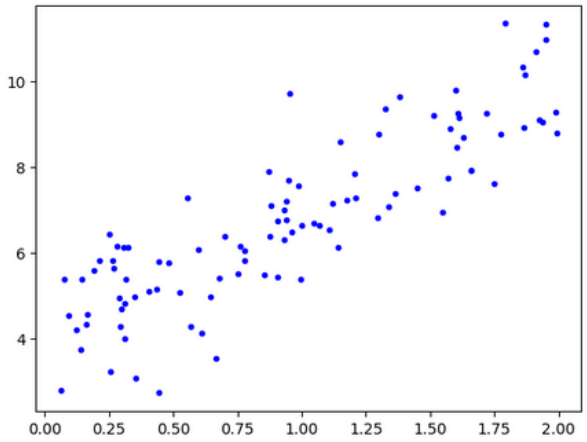
[1]:
```
import numpy as np

X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)

print("La longuitud del DataSet es:", len(X))
```

```
La longuitud del DataSet es: 100
```

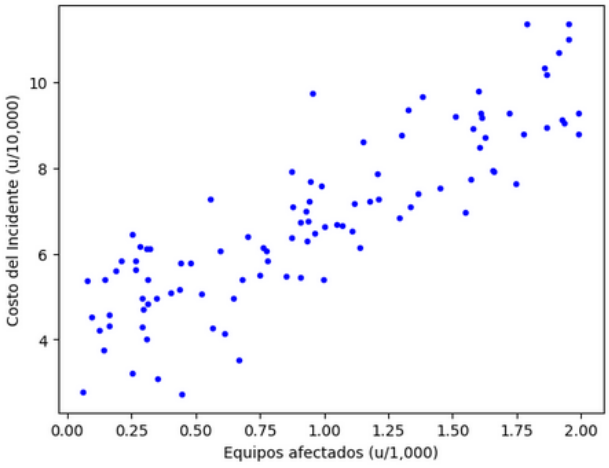
## 2.- Visualización del DataSet

[2]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

[3]:
```
plt.plot(X, y, "b.")
plt.show()
```



[4]:
```
plt.plot(X, y, "b.")
plt.xlabel("Equipos afectados (u/1,000)")
plt.ylabel("Costo del Incidente (u/10,000)")
plt.show()
```




## 3.- Modificación del DataSet

[5]:
```
import pandas as pd
```

[6]:
```
data = {'No_Equipos_Afectados': X.flatten(), 'Costo': y.flatten()}
df = pd.DataFrame(data)
df.head(10)
```

[6]:

| | No_Equipos_Afectados | Costo |
|---|---|---|
| 0 | 1.207574 | 7.851558 |
| 1 | 1.210680 | 7.286163 |
| 2 | 1.600521 | 9.802123 |
| 3 | 0.931868 | 6.306279 |
| 4 | 0.322858 | 6.126827 |
| 5 | 0.566703 | 4.272242 |
| 6 | 1.570028 | 7.745946 |
| 7 | 0.141568 | 3.755650 |
| 8 | 1.118397 | 7.161252 |
| 9 | 1.719318 | 9.269371 |

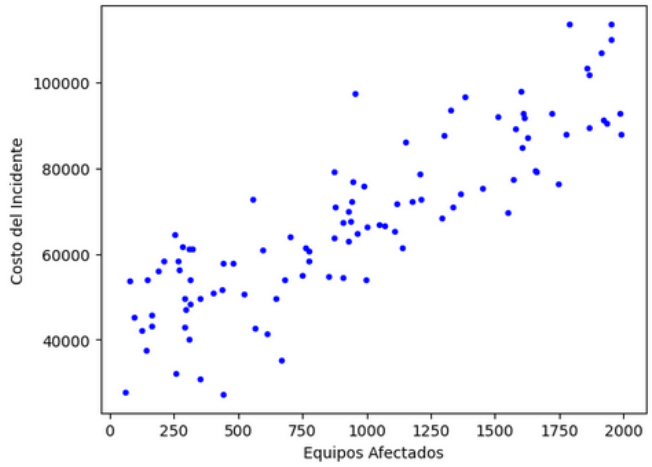
[7]:
```
# Escalado del número de equipos afectados
df['No_Equipos_Afectados'] = df['No_Equipos_Afectados'] * 1000
```

[7]:
```
# Escalado del número de equipos afectados
df['No_Equipos_Afectados'] = df['No_Equipos_Afectados'] * 1000
df['No_Equipos_Afectados'] = df['No_Equipos_Afectados'].astype('int')

#Escalado del costo
df['Costo'] = df['Costo'] * 10000
df['Costo'] = df['Costo'].astype('int')
df.head(10)
```

[7]:

| | No_Equipos_Afectados | Costo |
|---|---|---|
| 0 | 1207 | 78515 |
| 1 | 1210 | 72861 |
| 2 | 1600 | 98021 |
| 3 | 931 | 63062 |
| 4 | 322 | 61268 |
| 5 | 566 | 42722 |
| 6 | 1570 | 77459 |
| 7 | 141 | 37556 |
| 8 | 1118 | 71612 |
| 9 | 1719 | 92693 |

[8]:
```
plt.plot(df['No_Equipos_Afectados'], df['Costo'], "b.")
plt.xlabel("Equipos Afectados")
plt.ylabel("Costo del Incidente")
plt.show()
```




## 4.- Construcción del Modelo.

## 4.- Construcción del Modelo.

[9]:
```
from sklearn.linear_model import LinearRegression
```

[10]:
```
# Construcción del modelo y ajuste de la función de hipótesis
lin_reg = LinearRegression()
lin_reg.fit(df['No_Equipos_Afectados'].values.reshape(-1,1),df['Costo'].values)
```

[10]:
```
LinearRegression()
```

[11]:
```
# Parámetro Tetha 0
lin_reg.intercept_
```

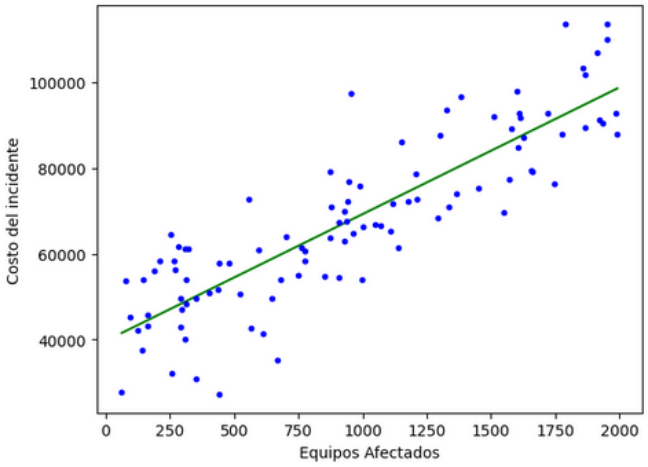
[11]:
```
np.float64(39712.47415191028)
```

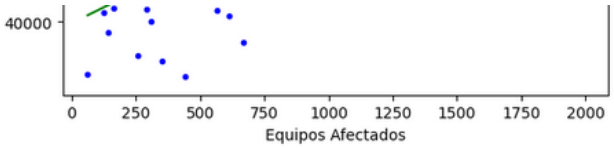
[12]:
```
# Parámetro Tetha 1
lin_reg.coef_
```

[12]:
```
array([29.53443486])
```

[13]:
```
# Predicción para el valor minimo y máximo para el conjunto de datos de entrenamiento.
X_min_max = np.array([[df["No_Equipos_Afectados"].min()],[df["No_Equipos_Afectados"].max()]])
y_train_pred = lin_reg.predict(X_min_max)
```

[15]:
```
# REpresentacion grafica de la función de hipotesis generada
plt.plot(X_min_max, y_train_pred, "g-")
plt.plot(df['No_Equipos_Afectados'], df['Costo'], "b.")
plt.xlabel("Equipos Afectados")
plt.ylabel("Costo del incidente")
plt.show()
```



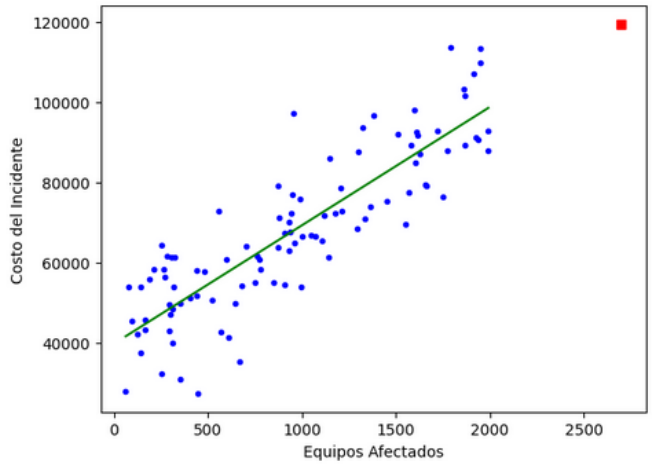



## 5.- Predicción de nuevos ejemplos

[21]:
```
x_new = np.array([[2700]]) # Número de quipos afectados
# Predicción del costo que tendria el incidente.
Costo = lin_reg.predict(x_new)
print("El costo del incidente sería: $ ", int(Costo[0]))
```

```
El costo del incidente sería: $  119455
```

[22]:
```
plt.plot(df['No_Equipos_Afectados'], df['Costo'], "b.")
plt.plot(X_min_max, y_train_pred, "g-")
plt.plot(x_new, Costo, "rs")
plt.xlabel("Equipos Afectados")
plt.ylabel("Costo del Incidente")
plt.show()
```




[ ]:

1. **Generación del DataSet**: Se crea un conjunto de datos con la cantidad de equipos afectados y el costo asociado.
2. **Visualización y Modificación del DataSet**: Se grafican los datos para observar su comportamiento y, si es necesario, se ajustan para mejorar el modelo.
3. **Construcción del Modelo**: Se utiliza la biblioteca sklearn para crear y ajustar un modelo de **regresión lineal**. El modelo genera una línea que predice el costo según el número de equipos afectados.
4. **Representación gráfica**: Se muestra una gráfica donde se compara la línea de predicción generada por el modelo con los datos reales.
5. **Predicción de nuevos ejemplos**: El modelo se utiliza para predecir el costo en función de diferentes cantidades de equipos afectados.

## Conclusión

Este ejercicio demuestra cómo la regresión lineal puede ser una herramienta útil para predecir el costo de un incidente de seguridad en función del número de equipos afectados. A través de un modelo matemático simple, es posible estimar cómo aumenta el costo a medida que más equipos se ven comprometidos. Al ajustar el modelo con datos reales y generar una gráfica, se puede visualizar la relación entre ambos factores y usarla para hacer predicciones sobre futuros incidentes de manera rápida y efectiva.