Regrecion lineal

Para presentar un regresion lineal realizaremos un ejemplo que representa el eje de las x como la edad de los niños y el eje de las y representa la cantidad de inversion de dineros segun la edad que tiene un niño Para comprobar vamos a realizar un programa en python donde podemos verificar como se relaisa la regresion lineal

Paso 1:

Vamos a imortar las librerias que necesitamos en python

In [32]:
```
import matplotlib.pyplot as plt
import numpy as np
import math as mt
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
```

Paso 2:

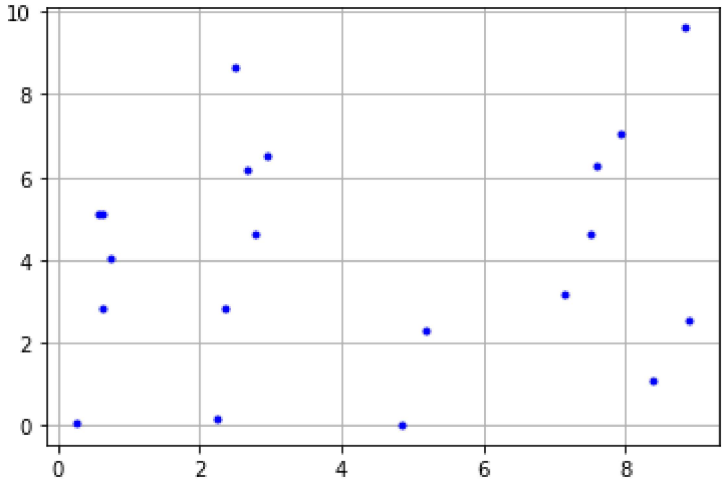
Vamos definir los puntos que vamos a necesitar para poder evaluar nuestra regresion lineal lo cual vamos a generar puntos que nos ayude a esto

In [33]:
```
def generar_datos():
    x = np.random.random(20)*10
    y = np.random.random(20)*10
    return x.reshape((20,1)), y.reshape((20,1))
X=np.array(generar_datos()[0])
Y=np.array(generar_datos()[1])
```

Paso 3:

Vamos a graficar todos los puntos que generamos a ver como estas en nuestra grafica

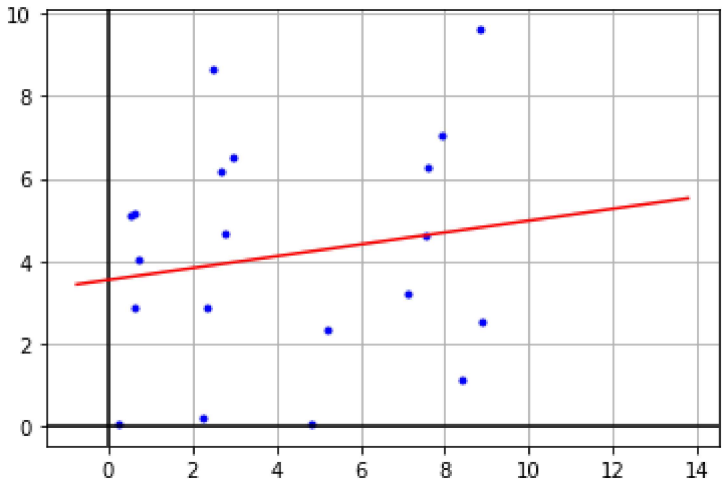
In [41]:
```
plt.plot(X,Y,'.',color="blue")
plt.grid(True)
plt.show()
ex=sum(X)
ey=sum(Y)
exy=sum(X*Y)
exx=sum(X*X)
lon=len(X)
```



In [42]:
```
m=(lon*exy-sum(X)*sum(Y))/(lon*exx-mt.pow(abs(ex),2))
b=(ey*exx-ex*exy)/(lon*exx-mt.pow(abs(ex),2))
print('dato',exy)
ecua=""
m=round(m[0],4)
b=round(b[0],4)
if (b < 0):
    ecua='y = {}x {}'
else:
    ecua='y = {}x + {}'
print(ecua.format(-1*m,b))
```

```
dato [378.37323205]
y = -0.144x + 3.5363
```

In [50]:
```
fu=lambda x: m*x+b
li=np.arange(min(X)-1.0,max(X)+5.0,0.5)
plt.plot(X,Y,'.',color="blue")
plt.axhline(y=0,color="black")
plt.axvline(x=0,color="black")
plt.plot(li,fu(li),color="red")
plt.grid(True)
plt.show()
```



Regresión Lineal con scikit-learn

Vamos a generar valor de 0 a 50 en los intervalos de 0.4 y vamso a graficar para ver como estas

nuestros puntos generado

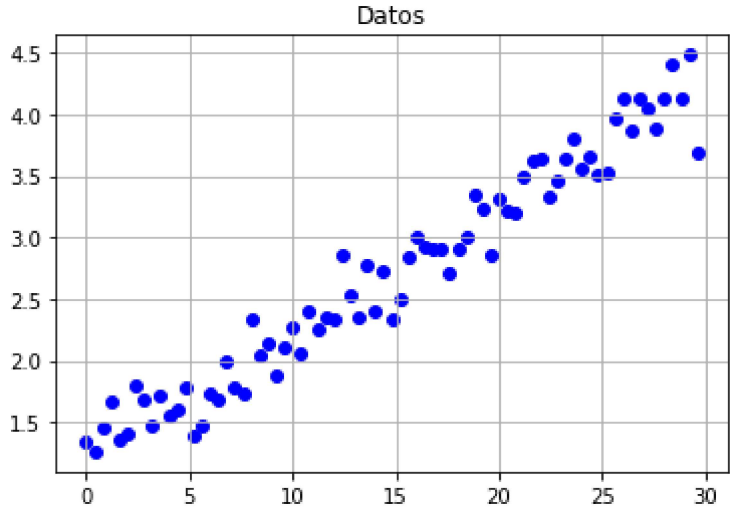
In [58]:
```
def f(x):
    np.random.seed(42)
    y = 0.1*x + 1.25 + 0.2*np.random.randn(x.shape[0])
    return y
```

Generamos numeros de 0 a 30 en interalos de 0.4

In [73]:
```
x = np.arange(0, 30 , 0.4)
y = f(x)
```

Grafica

In [74]:
```
plt.scatter(x,y,label='data', color='blue')
plt.grid(True)
plt.title('Datos');
```



Creamos una instancia de LinearRegression

In [75]:
```
regresion_lineal = LinearRegression()
```

Ingresamos los dato(x,y) a nestra regresion lineal

In [76]:
```
regresion_lineal.fit(x.reshape(-1,1), y)
```

Out[76]:
```
LinearRegression()
```

Revisamos los parametros

In [77]:
```
print('w = ' + str(regresion_lineal.coef_) + ', b = ' + str(regresion_lineal.intercept_
```

```
w = [0.10118582], b = 1.209459534633981
```

In [78]:
```
fu=lambda x: regresion_lineal.coef_*x+regresion_lineal.intercept_
li=np.arange(min(x)-5.0,max(x)+5.0,0.5)
```

Graficamos nuestra regresion lineal

In [79]:
```
plt.scatter(x,y,label='data', color='blue')
plt.plot(li,fu(li),color="red")
plt.grid(True)
plt.show()
```