Nombre: Jorge Arévalo

Docente: Ing. Diego Quisi

Materia: Simulación

REGESION LINEAL

In [36]:

```
import numpy as np
import random
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
%matplotlib inline

```

In [37]:

```
# Generador de distribución de datos para regresión lineal simple
def generador_datos_simple(beta, muestras, desviacion):
# Genero n (muestras) valores de x aleatorios entre 0 y 100
    x = np.random.random(muestras) * 100
# Genero un error aleatorio gaussiano con desviación típica (desviacion)
    e = np.random.randn(muestras) * desviacion
# Obtengo el y real como x*beta + error
    y = x * beta + e
    return x.reshape((muestras,1)), y.reshape((muestras,1))

```

In [43]:

```
# Parámetros de la distribución
desviacion = 200
beta = 10
n = 50
x, y = generador_datos_simple(beta, n, desviacion)
print('Datos en X:',x)
print('Datos en Y:',y)
```

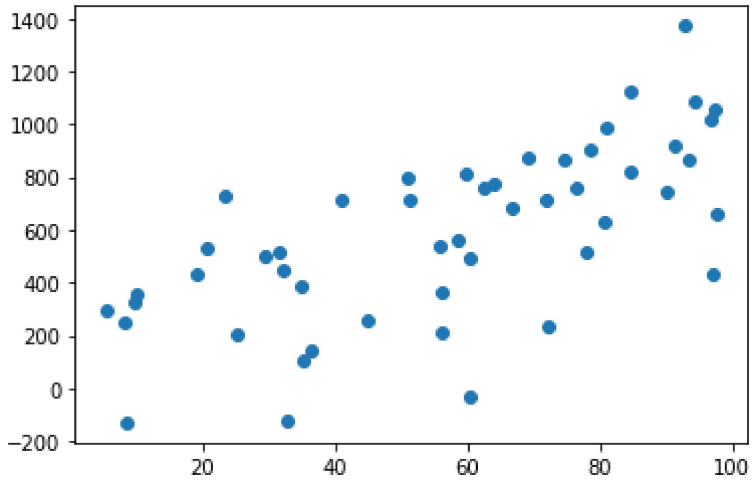
```
Datos en X: [[44.91288826]
 [82.30395836]
 [58.29539527]
 [40.58721592]
 [30.93105732]
 [98.81748645]
 [19.35627593]
 [72.35250848]
 [60.38276856]
 [13.03075465]
 [76.77210683]
 [24.89638097]
 [61.18005996]
 [ 8.45011698]
 [38.56447357]
 [77.44326889]
 [27.8872075 ]
 [70.8695727 ]
 [65.34632199]
 [12.90286206]
 [72.6620095 ]
 [76.90926054]
 [19.81479127]
 [62.05090138]
 [95.9187143 ]
 [ 4.56228532]
 [ 1.88160931]
 [ 6.62787421]
 [41.49511413]
 [ 2.52654166]
 [32.70465605]
 [ 8.56434888]
 [74.13361337]
 [74.05935032]
 [10.13079302]
 [51.41083397]
 [50.82729727]
 [80.65312877]
 [53.17497356]
 [81.31229518]
 [ 9.07797807]
 [18.64701846]
 [25.13711991]
 [87.84611229]
 [96.57979064]
 [19.48309584]
 [ 2.96616018]
 [27.88893841]
 [21.08646798]
 [33.70436481]]
Datos en Y: [[ 509.23170814]
```

```
[ 793.15985488]
[ 768.93564387]
[ 384.10464493]
[ 125.60904219]
[ 892.65841346]
[  80.24579104]
[ 711.25865055]
[ 659.82663121]
[ -99.94912274]
[ 919.79572091]
[ 594.21605152]
[ 872.78485601]
[-235.6254177 ]
[ 468.77224523]
[ 875.40249712]
[  60.40472948]
[ 357.5686108 ]
[ 525.96923879]
[ 263.97403018]
[ 400.15031053]
[ 353.91286291]
[ 277.13021937]
[ 785.85749829]
[1057.92106267]
[ -82.11387211]
[-120.02491297]
[ -26.13086175]
[ 614.81808914]
[ 152.83407715]
[ 499.78092293]
[ 102.86204477]
[ 879.22189549]
[ 976.67062   ]
[  -5.3985307 ]
[ 574.54884925]
[ 281.58990561]
[ 543.59187891]
[ 628.44098384]
[ 791.70644438]
[ 114.10533045]
[  74.01343406]
[ 372.97047279]
[1161.44204856]
[ 616.77532454]
[ 242.70923053]
[ 307.27729361]
[ 756.10718363]
[ 405.79927587]
[ 443.93957805]]
```

In [39]:

```
# Represento los datos generados
plt.scatter(x, y)
plt.show()

```



In [40]:

```
# Creo un modelo de regresión lineal
modelo = linear_model.LinearRegression()

# Entreno el modelo con los datos (X,Y)
modelo.fit(x, y)
# Ahora puedo obtener el coeficiente b_1
print(u'Coeficiente de determinación: ', modelo.coef_[0])

# Podemos predecir usando el modelo
y_pred = modelo.predict(x)

# Por último, calculamos el error cuadrático medio y el estadístico R^2
print(u'Error cuadrático medio: %.2f' % mean_squared_error(y, y_pred))
print(u'Estadístico R_2: %.2f' % r2_score(y, y_pred))

```

```
Coeficiente beta1:  [7.89148096]
Error cuadrático medio: 59266.78
Estadístico R_2: 0.45
```

In [41]:

```
# Representamos el ajuste (rojo) y la recta Y = beta*x (verde)
plt.scatter(x, y)
plt.plot(x, y_pred, color='red')
x_real = np.array([0, 100])
y_real = x_real*beta
plt.plot(x_real, y_real, color='green')
plt.show()

```