Nombre: ***Macias Pico Josselyn Stefany***

Curso: ***Sexto "B"***

Materia: ***Modelamiento y simulacion***

Docente: ***Ing. Jorge Anibal Moya Delgado***

1. La relación que existe entre el porcentaje de desempleo y el grado de criminalidad en el país se muestra en la tabla:

a. Genere la ecuación cuadrática con los datos del problema

In [5]:
```
# Importar libreria numpy
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# datos experimentales
# el DataFrame se llama movil
exporta = {'% de desempleo':[7.2 ,10.4, 9.7, 6.8, 8.0, 5.9, 4.8, 6.6, 5.6],
 'Robos y asaltos por cada 10,000 habitantes':[7.4, 8.6, 6.6, 5.5, 6.9, 3.5, 2.4,
a = pd.DataFrame(exporta)
x = a.index.values
y= a["Robos y asaltos por cada 10,000 habitantes"]
p = np.polyfit(x,y,2)
p1,p2,p3 = p
print ("El valor de p0 = ", p1, "Valor de p1 = ", p2, " el valor de p2 = ",p3)
```

```
El valor de p0 =  0.04577922077922062 Valor de p1 =  -0.9045670995670978  el va
lor de p2 =  8.24727272727272
```

a. Calcule el pronóstico para la tasa de desempleo de 7.1%

In [16]:
```
n=x.size
x1 = []
x2 = []
for i in [12,13,14]:
    y1_ajuste = p[0]*i*i + p[1]*i + p[2]
    print (f" z = {i} w = {y1_ajuste}")
    x1.append(i)
    x2.append(y1_ajuste)
a["y_ajuste"]=y_ajuste
dp = pd.DataFrame({'Desempleo':[45,50,55], 'Robos':[0,0,0],'y_ajuste':x2})
dp
a = a.append(dp,ignore_index=True)
a
```

```
 z = 12 w = 3.984675324675317
 z = 13 w = 4.2245887445887345
 z = 14 w = 4.556060606060592

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-16-a6db8f509e2a> in <module>
      7     x1.append(i)
      8     x2.append(y1_ajuste)
----> 9 a["y_ajuste"]=y_ajuste
     10 dp = pd.DataFrame({'Desempleo':[45,50,55], 'Robos':[0,0,0],'y_ajuste':x
2})
     11 dp

NameError: name 'y_ajuste' is not defined
```

2. Servicio de consutaria a estudiantes de la universidad

a) Utilice la técnica del promedio móvil a tres periodos con os datos del problema

In [7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# el DataFrame se llama movil
exporta = {'Semanas':[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
 'Estudiantes':[688, 745, 780, 790, 1050, 870, 650, 670, 750, 794, 820, 1120]}
movil = pd.DataFrame(exporta)
# mostramos los 5 primeros registros
movil.head()
# calculamos para la primera media móvil MMO_3
for i in range(0,movil.shape[0]-2):
 movil.loc[movil.index[i+2],'MMO_3'] = np.round(((movil.iloc[i,1]+movil.iloc[i+1,
# calculamos para la segunda media móvil MMO_4
for i in range(0,movil.shape[0]-3):
 movil.loc[movil.index[i+3],'MMO_4'] = np.round(((movil.iloc[i,1]+movil.iloc[i+1,
3,1])/4),1)
# calculamos la proyeción final
proyeccion = movil.iloc[7:,[1,2,3]]
p1,p2,p3 =proyeccion.mean()
# incorporamos al DataFrame
a = movil.append({'Semanas':7.1,'Estudiantes':p1, 'MMO_3':p2, 'MMO_4':p3},ignore_
# mostramos los resultados
a['e_MM3'] = a['Estudiantes']-a['MMO_3']
a['e_MM4'] = a['Estudiantes']-a['MMO_4']
a
```

Out[7]:

	Semanas	Estudiantes	MMO_3	MMO_4	e_MM3	e_MM4
0	1.0	688.0	NaN	NaN	NaN	NaN
1	2.0	745.0	NaN	NaN	NaN	NaN
2	3.0	780.0	737.70	NaN	42.30	NaN
3	4.0	790.0	771.70	750.8	18.30	39.2
4	5.0	1050.0	873.30	841.2	176.70	208.8
5	6.0	870.0	903.30	872.5	-33.30	-2.5
6	7.0	650.0	856.70	840.0	-206.70	-190.0
7	8.0	670.0	730.00	810.0	-60.00	-140.0
8	9.0	750.0	690.00	735.0	60.00	15.0
9	10.0	794.0	738.00	716.0	56.00	78.0
10	11.0	820.0	788.00	758.5	32.00	61.5
11	12.0	1120.0	911.30	871.0	208.70	249.0
12	7.1	830.8	771.46	778.1	59.34	52.7

Calcule el pronóstico para el mes siguiente

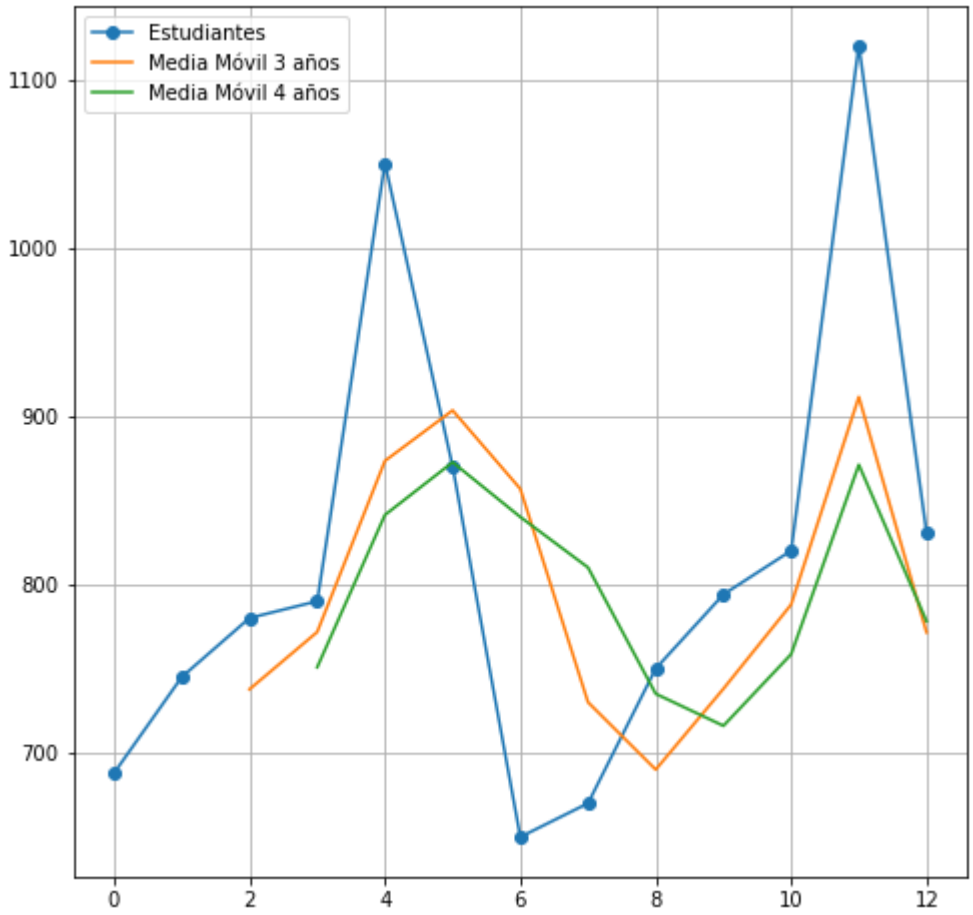
In [8]:
```
mos el promedio de los cada una de las columnas de df
m4,m5,m6 =a.mean()
ror Media Móvil 1 = ',round(m1),'Error Media Móvil 2 = ',round(m2), 'Error Media
```

```
Error Media Móvil 1 =  7 Error Media Móvil 2 =  812 Error Media Móvil 3 =  797
```

Grafica del mes siguiente

In [9]:
```
#%matplotlib inline
plt.figure(figsize=[8,8])
plt.grid(True)
plt.plot(a['Estudiantes'],label='Estudiantes',marker='o')
plt.plot(a['MMO_3'],label='Media Móvil 3 años')
plt.plot(a['MMO_4'],label='Media Móvil 4 años')
plt.legend(loc=2)
```

Out[9]: <matplotlib.legend.Legend at 0x284156984f0>



b) Utilice el modelo de suavizamiento exponencial con alfa = 0.4 con os datos del problema y Calcule el pronóstico para el mes siguiente

In [10]:
```
 ### Alisamiento Exponencial
# Vamos a crear un DataFrame con los datos y luego procederemos a calcular el pro
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# el DataFrame se llama movil
exporta = {'Semanas':[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
 'Estudiantes':[688, 745, 780, 790, 1050, 870, 650, 670, 750, 794, 820, 1120]}
movil = pd.DataFrame(exporta)
# mostramos los 5 primeros registros
movil.head()
alfa = 0.4
unoalfa = 1. - alfa
for i in range(0,movil.shape[0]-1):
    movil.loc[movil.index[i+1],'SN'] = np.round(movil.iloc[i,1],1)
for i in range(2,movil.shape[0]):
    movil.loc[movil.index[i],'SN'] = np.round(movil.iloc[i-1,1],1)*alfa + np.rour
i=i+1
p1=0
p2=np.round(movil.iloc[i-1,1],1)*alfa + np.round(movil.iloc[i-1,2],1)*unoalfa
a = movil.append({'Semanas':2018,'Estudiantes':p1, 'SN':p2},ignore_index=True)
a
```

Out[10]:

	Semanas	Estudiantes	SN
0	1.0	688.0	NaN
1	2.0	745.0	688.00
2	3.0	780.0	710.80
3	4.0	790.0	738.48
4	5.0	1050.0	759.10
5	6.0	870.0	875.46
6	7.0	650.0	873.30
7	8.0	670.0	783.98
8	9.0	750.0	738.40
9	10.0	794.0	743.04
10	11.0	820.0	763.40
11	12.0	1120.0	786.04
12	2018.0	0.0	919.60

3. Monto de pago inicial de hipotecas y número de hipotecas perdidas

a. Genere la ecuación lineal y luego la cuadrática con los datos del problema

In [29]:
```
# Importar libreria numpy
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# datos experimentales
# el DataFrame se llama movil
exporta = {'Monto de pagoinicial %':[5, 10, 15, 20, 25, 30, 35, 40],
 '# de Hipotecas Perdidas':[120, 95, 75, 45, 30, 20, 15, 10]}
a = pd.DataFrame(exporta)
x = a.index.values
y= a["# de Hipotecas Perdidas"]
p = np.polyfit(x,y,2)
p1,p2,p3 = p
print ("El valor de p0 = ", p1, "Valor de p1 = ", p2, " el valor de p2 = ",p3)
y_ajuste = p[0]*x*x + p[1]*x + p[2]
```

```
El valor de p0 =  2.1428571428571423 Valor de p1 =  -31.07142857142856  el valo
r de p2 =  122.49999999999997
```

Calcule el pronóstico para el porcentaje de pago del 23%; además indique cual es la mejor opción para el resultado.

In [30]:
```
n=x.size
x1 = []
x2 = []
for i in [12,13,14]:
    y1_ajuste = p[0]*i*i + p[1]*i + p[2]
    print (f" z = {i} w = {y1_ajuste}")
    x1.append(i)
    x2.append(y1_ajuste)
a["y_ajuste"]=y_ajuste
dp = pd.DataFrame({'PAGO INICIAL':[45,50,55], 'HIPOTECAS':[0,0,0],'y_ajuste':x2})
dp
a = a.append(dp,ignore_index=True)
a
```

```
 z = 12 w = 58.21428571428575
 z = 13 w = 80.71428571428575
 z = 14 w = 107.50000000000003
```

Out[30]:

	Monto de pagoinicial %	# de Hipotecas Perdidas	y_ajuste	PAGO INICIAL	HIPOTECAS
0	5.0	120.0	122.500000	NaN	NaN
1	10.0	95.0	93.571429	NaN	NaN
2	15.0	75.0	68.928571	NaN	NaN
3	20.0	45.0	48.571429	NaN	NaN
4	25.0	30.0	32.500000	NaN	NaN
5	30.0	20.0	20.714286	NaN	NaN
6	35.0	15.0	13.214286	NaN	NaN
7	40.0	10.0	10.000000	NaN	NaN
8	NaN	NaN	58.214286	45.0	0.0
9	NaN	NaN	80.714286	50.0	0.0
10	NaN	NaN	107.500000	55.0	0.0

In [ ]:
```
```