Ejercicios 1

José Ignacio Contreras Ruiz 1887848

Kevin Franco Gonzalez Gonzalez 1805425

Rafael Hernan Elizondo Aranda 1887942

Andre Damián Valdez Vázquez 1887881

Daniela Monserrat Sanabria Martinez 1563836

Regresión lineal

In [35]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Ejercicio #1 Regresion Lineal

In [12]:

```
y =[68.78,74.11,71.73,69.88,67.25,68.78,68.34,67.01,63.45,71.19,67.19,65.80,64.30,67.97,72.18,65.27,66.09,67.51,70.1,68.25,67.89,6
8.14,69.08,72.80,67.42,68.49,68.61,74.03,71.52,68.18]
x=[162,212,220,206,152,183,167,175,156,186,183,163,163,172,194,168,161,164,188,187,162,192,184,206,175,154,187,212,195,205]
```

In [32]:

```
x=np.array(x)
y=np.array(y)
x
y

n = len(x)
n
```

Out[32]:

```
30
```

In [33]:

```
sumx = sum(x)
sumy = sum(y)
sumx2 = sum(x**2)
sumy2 = sum(y**2)
sumxy = sum(x*y)

promx = sumx/n
promy = sumy/n
```

In [34]:

```
Bo = (sumx*sumy-n*sumxy)/(sumx**2-n*sumx2)
B1 = promy-Bo*promx

Bo,B1
#y=Bo*X+B1
```

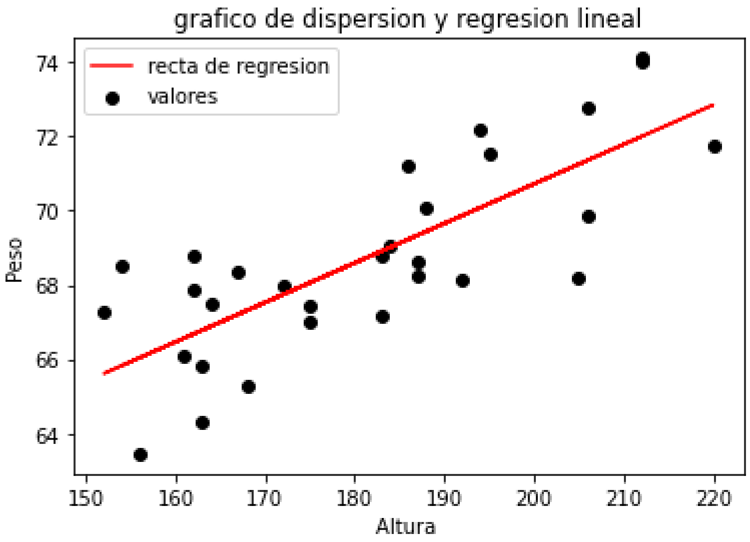
Out[34]:

```
(0.1063780544087021, 49.442721744770424)
```

La funcion lineal se obtiene de la ecuacion y=0.1063780544087021*X+49.442721744770424

In [31]:

```
import matplotlib.pyplot as plt
plt.scatter(x,y, c="black")
plt.plot(x,x*Bo+B1, c="red")
plt.title("grafico de dispersion y regresion lineal")
plt.xlabel("Altura ")
plt.ylabel("Peso")
plt.legend(["recta de regresion","valores"])
plt.show()
```



In [ ]:

```

```

1. Observa la tabla que se describe a continuación. Utilizando el agoritmo a priori, y la técnica de asociación, realiza la tabla de relaciones y resuelve cuál es el nivel K de soporte más alto al que podemos llegar teniendo un umbral de 0.5.

In [100]:

```
pip install apyori
```

```
Requirement already satisfied: apyori in c:\users\sofpr\anaconda3\lib\site-packages (1.1.2)
Note: you may need to restart the kernel to use updated packages.
```

In [101]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from apyori import apriori
```

In [102]:

```
datos = np.array((["A","B","C","E"], ["B","E"], ["C","D","E"],["A","C","D"],["A","C","E"]))
```

In [103]:

```
reglas = apriori(datos, min_support = 0.5, min_confidence = 0, min_lift = 0, min_length = 1)
```

In [104]:

```
resultados = list(reglas)
```

In [105]:

```
resultados
```

Out[105]:

```
[RelationRecord(items=frozenset({'A'}), support=0.6, ordered_statistics=[OrderedStatistic(items_base=frozenset(), item
s_add=frozenset({'A'}), confidence=0.6, lift=1.0)]),
 RelationRecord(items=frozenset({'C'}), support=0.8, ordered_statistics=[OrderedStatistic(items_base=frozenset(), item
s_add=frozenset({'C'}), confidence=0.8, lift=1.0)]),
 RelationRecord(items=frozenset({'E'}), support=0.8, ordered_statistics=[OrderedStatistic(items_base=frozenset(), item
s_add=frozenset({'E'}), confidence=0.8, lift=1.0)]),
 RelationRecord(items=frozenset({'A', 'C'}), support=0.6, ordered_statistics=[OrderedStatistic(items_base=frozenset(),
items_add=frozenset({'A', 'C'}), confidence=0.6, lift=1.0), OrderedStatistic(items_base=frozenset({'A'}), items_add=fr
ozenset({'C'}), confidence=1.0, lift=1.25), OrderedStatistic(items_base=frozenset({'C'}), items_add=frozenset({'A'}),
confidence=0.7499999999999999, lift=1.2499999999999998)]),
 RelationRecord(items=frozenset({'E', 'C'}), support=0.6, ordered_statistics=[OrderedStatistic(items_base=frozenset(),
items_add=frozenset({'E', 'C'}), confidence=0.6, lift=1.0), OrderedStatistic(items_base=frozenset({'C'}), items_add=fr
ozenset({'E'}), confidence=0.7499999999999999, lift=0.9374999999999998), OrderedStatistic(items_base=frozenset({'E'}),
items_add=frozenset({'C'}), confidence=0.7499999999999999, lift=0.9374999999999998)])]
```

In [106]:

```python
def inspect(resultados):
    rh          = [tuple(resultado[2][0][0]) for resultado in resultados]
    lh          = [tuple(resultado[2][0][1]) for resultado in resultados]
    soporte    = [resultado[1] for resultado in resultados]
    confianza = [resultado[2][0][2] for resultado in resultados]
    lift       = [resultado[2][0][3] for resultado in resultados]
    return list(zip(rh, lh, soporte, confianza, lift))
```

In [107]:

```python
resultadoDataFrame = pd.DataFrame(inspect(resultados),
                columns=['rhs','lhs','soporte','confianza','lift'])
```

In [108]:

```python
resultadoDataFrame
```

Out[108]:

	rhs	lhs	soporte	confianza	lift
0	()	(A,)	0.6	0.6	1.0
1	()	(C,)	0.8	0.8	1.0
2	()	(E,)	0.8	0.8	1.0
3	()	(A, C)	0.6	0.6	1.0
4	()	(E, C)	0.6	0.6	1.0

Conclusión

Los niveles de K de soporte más alto al que podemos llegar con estos datos teniendo un soporte mínimo de 0.5 es:

```
Cuando K=1
(A)
Soporte: 0.6
Confianza: 0.6
Lift: 1.0
(C)
Soporte: 0.8
Confianza: 0.8
Lift: 1.0
(E)
Soporte: 0.8
Confianza: 0.8
Lift: 1.0
```

Cuando K=2
(C,A)
Soporte: 0.6
Confianza: 0.6
Lift: 1.0

(A->C)
Soporte: 0.6
Confianza: 1.0
Lift: 1.25
(C->A)
Soporte: 0.6
Confianza: 0.0.7499999999999999
Lift: 1.2499999999999998

(E,C)
Soporte: 0.6
Confianza: 0.6
Lift: 1.0

(E->C) Support: 0.6
Confidence: 0.7499999999999999
Lift:0.9374999999999998
(C->E)
Support: 0.6
Confidence: 0.7499999999999999
Lift: 0.9374999999999998