In [1]:

```
# Importando as bibliotecas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sbs
%matplotlib inline
from statsmodels.formula.api import ols
```

In [2]:

```
# Carregando os dados
data = pd.read_csv ('pesos2.csv')
```

In [3]:

```
# Verificando as 5 primeiras linhas
data.head ( )
```

Out[3]:

	Sexo	Grupo	Head Size	Brain Weight
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590

In [4]:

```
# Verificando as 15 primeiras linhas
data.head (15)
```

Out[4]:

	Sexo	Grupo	Head Size	Brain Weight
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590
5	1	1	3585	1300
6	1	1	3785	1400
7	1	1	3559	1255
8	1	1	3613	1355
9	1	1	3982	1375
10	1	1	3443	1340
11	1	1	3993	1380
12	1	1	3640	1355
13	1	1	4208	1522
14	1	1	3832	1208

In [5]:

```
# Verificando as 5 últimas linhas
data.tail ( )
```

Out[5]:

	Sexo	Grupo	Head Size	Brain Weight
232	2	2	3214	1110
233	2	2	3394	1215
234	2	2	3233	1104
235	2	2	3352	1170
236	2	2	3391	1120

In [6]:

```
#Verificando o número de linhas e colunas
data.shape
```

Out[6]:

```
(237, 4)
```

In [7]:

```
#Verificar se há valores NAN (Retirar se houver)
data.isnull().sum()
```

Out[7]:

```
Sexo           0
Grupo          0
Head Size      0
Brain Weight   0
dtype: int64
```

In [8]:

```
#Retirando a coluna Grupo
data.drop('Grupo', axis = 1, inplace=True)
```

In [9]:

```
data.shape
```

Out[9]:

```
(237, 3)
```

In [10]:

```
data.head( )
```

Out[10]:

	Sexo	Head Size	Brain Weight
0	1	4512	1530
1	1	3738	1297
2	1	4261	1335
3	1	3777	1282
4	1	4177	1590

In [11]:

```
#Utilizar o método describe para conhecer sua tabela
data.describe( )
```

Out[11]:

	Sexo	Head Size	Brain Weight
count	237.000000	237.000000	237.000000
mean	1.434599	3633.991561	1282.873418
std	0.496753	365.261422	120.340446
min	1.000000	2720.000000	955.000000
25%	1.000000	3389.000000	1207.000000
50%	1.000000	3614.000000	1280.000000
75%	2.000000	3876.000000	1350.000000
max	2.000000	4747.000000	1635.000000

In [16]:

```
data.rename(columns={'Head Size':'cabeca', 'Brain Weight':'peso'}, inplace= True)
```

In [18]:

```
data.corr()
```

Out[18]:

	Sexo	cabeca	peso
Sexo	1.000000	-0.51405	-0.465266
cabeca	-0.514050	1.00000	0.799570
peso	-0.465266	0.79957	1.000000

In [19]:

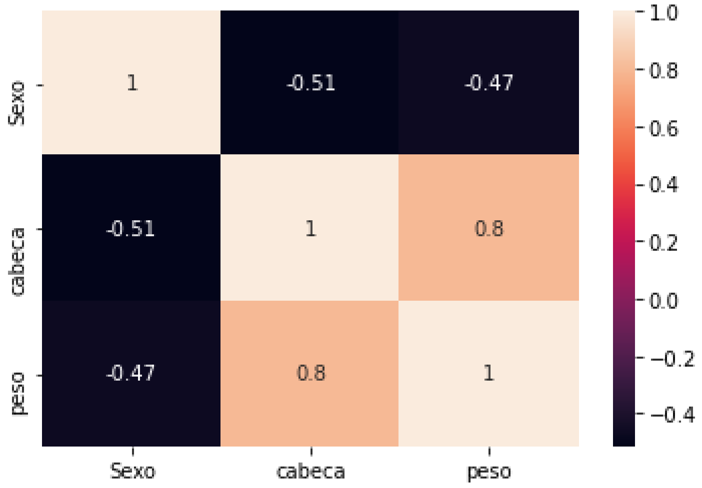
```
corr= data.corr()
```

In [20]:

```
sbs.heatmap(corr,annot = True)
```

Out[20]:

<matplotlib.axes._subplots.AxesSubplot at 0x5feadf88e0>

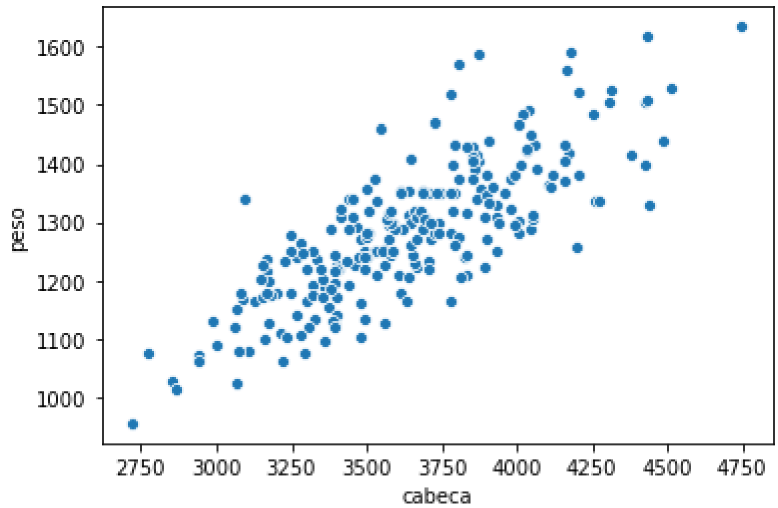


In [21]:

```
sbs.scatterplot(x='cabeca' , y='peso', data=data)
```

Out[21]:

<matplotlib.axes._subplots.AxesSubplot at 0x5ff18056a0>



In [22]:

```
formula = 'cabeca ~ peso'
modelo_v1 = ols (formula, data = data).fit()
modelo_v1.summary()
```

Out[22]:

OLS Regression Results

Dep. Variable:	cabeca	**R-squared:**	0.639
Model:	OLS	**Adj. R-squared:**	0.638
Method:	Least Squares	**F-statistic:**	416.5
Date:	Tue, 22 Sep 2020	**Prob (F-statistic):**	5.96e-54
Time:	21:57:55	**Log-Likelihood:**	-1613.4
No. Observations:	237	**AIC:**	3231.
Df Residuals:	235	**BIC:**	3238.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	520.6101	153.215	3.398	0.001	218.759	822.461
peso	2.4269	0.119	20.409	0.000	2.193	2.661

Omnibus:	2.687	**Durbin-Watson:**	1.726
Prob(Omnibus):	0.261	**Jarque-Bera (JB):**	2.321
Skew:	0.207	**Prob(JB):**	0.313
Kurtosis:	3.252	**Cond. No.**	1.38e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.38e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

In [24]:

```
sbs.lmplot(x='cabeca', y='peso', data=data,fit_reg=True)
```

Out[24]:

```
<seaborn.axisgrid.FacetGrid at 0x5ff18bc7c0>
```