

Nombre : Josue Alejandro Sauca Pucha

Fecha : 28-06-2023

```
In [1]: import pandas as pd                ## Este proporciona una estructura simi
import statsmodels.api as sm             ## Este proporciona funciones para la e
import statsmodels.formula.api as smf    ## Permite ajustar modelos estadísticos
```

```
In [11]: dt = pd.read_csv('procesamientode_datos.csv')
dt
```

```
Out[11]:
```

	Coste	Ficheros	Flujos	Procesos
0	22.6	4	44	18
1	15.0	2	33	15
2	78.1	20	80	80
3	28.0	6	24	21
4	80.5	6	227	50
5	24.5	3	20	18
6	20.5	4	41	13
7	147.6	16	187	137
8	4.2	4	19	15
9	48.2	6	50	21
10	20.5	5	48	17

```
In [12]: dt.corr()
```

```
Out[12]:
```

	Coste	Ficheros	Flujos	Procesos
Coste	1.000000	0.778474	0.830392	0.959842
Ficheros	0.778474	1.000000	0.458982	0.854561
Flujos	0.830392	0.458982	1.000000	0.720437
Procesos	0.959842	0.854561	0.720437	1.000000

```
In [13]: mod = smf.ols('Coste~Ficheros+Flujos+Procesos', data=dt).fit() # Ajusta el mo
```

```
In [14]: mod.summary()
```

```
D:\Programas\Anaconda\lib\site-packages\scipy\stats\stats.py:1541: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 ... continuing ")
```

```
Out[14]: OLS Regression Results
```

Dep. Variable:	Coste	R-squared:	0.961
Model:	OLS	Adj. R-squared:	0.945
Method:	Least Squares	F-statistic:	58.21
Date:	Wed, 28 Jun 2023	Prob (F-statistic):	2.58e-05
Time:	11:23:31	Log-Likelihood:	-38.361
No. Observations:	11	AIC:	84.72
Df Residuals:	7	BIC:	86.31
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
Intercept	1.9618	5.608	0.350	0.737	-11.300	15.223
Ficheros	0.1178	1.177	0.100	0.923	-2.665	2.900
Flujos	0.1767	0.071	2.474	0.043	0.008	0.346
Procesos	0.7964	0.220	3.613	0.009	0.275	1.318

Omnibus:	7.167	Durbin-Watson:	3.064
Prob(Omnibus):	0.028	Jarque-Bera (JB):	2.833
Skew:	0.987	Prob(JB):	0.243
Kurtosis:	4.513	Cond. No.	204.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
In [6]: mod.params
```

```
Out[6]: Intercept    1.961779
Ficheros           0.117759
Flujos             0.176726
Procesos           0.796448
dtype: float64
```

```
In [15]: mod.params.Ficheros
```

```
Out[15]: 0.11775864437820971
```

```
In [24]: def predicion(Ficheros, Flujos, Procesos):  
         prediccion = mod.params.Intercept + mod.params.Ficheros*Ficheros + mod.params.Flujos*Flujos + mod.params.Procesos*Procesos  
         return prediccion
```

```
In [25]: predicion(10,100,75)
```

```
Out[25]: 80.54557706520916
```