Previamente… Módulo 3 - Tarea 3

Análisis de datos.

Este tema describirá los procesos usados para analizar el contenido de los datos.

Repaso de los datos a procesar

Vamos a practicar el análisis de datos en una *Jupyter Notebook* con las librerías Pandas de un archivo Excel con información sobre producción de ganado en Europa que ha sido limpiada en el tema anterior.

Input [1]:

```
import pandas as pd
import numpy as np
file = 'datos/animalEurostatNuts2_corrected.xlsx'
data = pd.read_excel(file, sheet_name='Data', index_col=0)
data.head(5)
```

Output [1]:

NUTS	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	...	2010	2011	2012	2013	2014	
EU	86785.24	86785.24	86785.24	86785.24	86785.24	86785.24	86785.24	86785.24	86785.24	86785.24	...	86785.24	86785.24	86785.24	86785.24	86785.24	8
BE	3105.50	3099.60	3084.20	3161.10	3158.70	3070.80	2978.40	2984.40	2970.40	3041.60	...	2592.63	2560.32	2484.27	2432.53	2477.24	2
BE1	0.30	0.50	0.50	0.50	0.40	0.40	0.40	0.30	0.40	0.40	...	0.24	0.25	0.56	0.59	0.79	
BE10	0.30	0.50	0.50	0.50	0.40	0.40	0.40	0.30	0.40	0.40	...	0.24	0.25	0.56	0.59	0.79	
BE2	1655.20	1661.40	1637.20	1685.50	1678.50	1613.10	1556.80	1554.40	1536.20	1558.10	...	1303.87	1302.25	1269.41	1255.40	1299.98	

5 rows × 29 columns

Identificación de correlaciones

Identificar correspondencias entre conjuntos de datos numéricos es importante para determinar si hay columnas con información redundante (o para detectar dos columnas que deberían estar correlacionadas y no lo están)

- *dataframe.corr() muestral la correlación entre las columnas del conjunto de datos.*

Input [2]:

```
data.corr()
```

Output [2]:

	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	...	2010	2011	2012	2013	2014	
1991	1.000000	0.999852	0.999602	0.999482	0.999422	0.999278	0.999061	0.998814	0.998730	0.998267	...	0.993435	0.991994	0.990591	0.990137	0.990538	0.9
1992	0.999852	1.000000	0.999917	0.999826	0.999766	0.999670	0.999547	0.999362	0.999257	0.998899	...	0.994328	0.992934	0.991553	0.991070	0.991452	0.9
1993	0.999602	0.999917	1.000000	0.999940	0.999883	0.999815	0.999740	0.999590	0.999488	0.999167	...	0.994678	0.993306	0.991932	0.991433	0.991807	0.9
1994	0.999482	0.999826	0.999940	1.000000	0.999968	0.999903	0.999813	0.999655	0.999575	0.999329	...	0.994748	0.993326	0.991950	0.991455	0.991844	0.9
1995	0.999422	0.999766	0.999883	0.999968	1.000000	0.999945	0.999870	0.999708	0.999651	0.999403	...	0.994840	0.993407	0.992020	0.991526	0.991927	0.9
1996	0.999278	0.999670	0.999815	0.999903	0.999945	1.000000	0.999939	0.999839	0.999795	0.999577	...	0.995155	0.993751	0.992393	0.991895	0.992295	0.9
1997	0.999061	0.999547	0.999740	0.999813	0.999870	0.999939	1.000000	0.999915	0.999835	0.999611	...	0.995222	0.993821	0.992463	0.991942	0.992332	0.9
1998	0.998814	0.999362	0.999590	0.999655	0.999708	0.999839	0.999915	1.000000	0.999939	0.999746	...	0.995474	0.994102	0.992756	0.992223	0.992601	0.9
1999	0.998730	0.999257	0.999488	0.999575	0.999651	0.999795	0.999835	0.999939	1.000000	0.999851	...	0.995645	0.994261	0.992874	0.992332	0.992725	0.9
2000	0.998267	0.998899	0.999167	0.999329	0.999403	0.999577	0.999611	0.999746	0.999851	1.000000	...	0.995948	0.994552	0.993111	0.992538	0.992945	0.9
2001	0.998252	0.998828	0.999065	0.999182	0.999309	0.999529	0.999567	0.999726	0.999846	0.999904	...	0.995878	0.994501	0.993063	0.992500	0.992903	0.9
2002	0.997903	0.998596	0.998889	0.998975	0.999081	0.999342	0.999432	0.999641	0.999776	0.999877	...	0.996389	0.995107	0.993723	0.993154	0.993539	0.9
2003	0.997481	0.998240	0.998571	0.998613	0.998696	0.998970	0.999134	0.999370	0.999512	0.999644	...	0.994879	0.993373	0.991723	0.991035	0.991477	0.9
2004	0.997181	0.998000	0.998353	0.998378	0.998468	0.998775	0.998961	0.999227	0.999381	0.999521	...	0.996252	0.994968	0.993520	0.992898	0.993293	0.9
2005	0.997098	0.997948	0.998304	0.998327	0.998413	0.998727	0.998915	0.999178	0.999292	0.999414	...	0.997349	0.996266	0.995034	0.994490	0.994832	0.9
2006	0.996721	0.997583	0.997940	0.997963	0.998058	0.998390	0.998559	0.998831	0.998961	0.999090	...	0.998252	0.997368	0.996325	0.995853	0.996151	0.9
2007	0.996117	0.997000	0.997361	0.997389	0.997495	0.997841	0.997988	0.998270	0.998420	0.998562	...	0.998955	0.998269	0.997414	0.997014	0.997272	0.9
2008	0.759147	0.759985	0.761122	0.762156	0.760412	0.758565	0.759006	0.758570	0.759027	0.756592	...	0.750940	0.749026	0.748746	0.748106	0.748310	0.7
2009	0.994825	0.995674	0.996004	0.996087	0.996187	0.996487	0.996559	0.996798	0.996923	0.997160	...	0.999850	0.999513	0.999009	0.998771	0.998945	0.9
2010	0.993435	0.994328	0.994678	0.994748	0.994840	0.995155	0.995222	0.995474	0.995645	0.995948	...	1.000000	0.999876	0.999536	0.999356	0.999488	0.9
2011	0.991994	0.992934	0.993306	0.993326	0.993407	0.993751	0.993821	0.994102	0.994261	0.994552	...	0.999876	1.000000	0.999865	0.999754	0.999825	0.9
2012	0.990591	0.991553	0.991932	0.991950	0.992020	0.992393	0.992463	0.992756	0.992874	0.993111	...	0.999536	0.999865	1.000000	0.999973	0.999977	0.9
2013	0.990137	0.991070	0.991433	0.991455	0.991526	0.991895	0.991942	0.992223	0.992332	0.992538	...	0.999356	0.999754	0.999973	1.000000	0.999982	0.9
2014	0.990538	0.991452	0.991807	0.991844	0.991927	0.992295	0.992332	0.992601	0.992725	0.992945	...	0.999488	0.999825	0.999977	0.999982	1.000000	0.9
2015	0.990646	0.991569	0.991930	0.991987	0.992080	0.992456	0.992511	0.992774	0.992887	0.993089	...	0.999499	0.999805	0.999954	0.999955	0.999985	1.0
2016	0.990320	0.991291	0.991683	0.991713	0.991807	0.992215	0.992312	0.992585	0.992670	0.992822	...	0.999379	0.999729	0.999919	0.999925	0.999945	0.9
2017	0.987651	0.988655	0.989067	0.989105	0.989207	0.989652	0.989735	0.990023	0.990119	0.990267	...	0.998543	0.999181	0.999654	0.999762	0.999716	0.9
2018	0.985611	0.986680	0.987126	0.987139	0.987241	0.987717	0.987812	0.988122	0.988219	0.988378	...	0.997787	0.998628	0.999260	0.999417	0.999327	0.9
2019	0.985606	0.986676	0.987122	0.987134	0.987237	0.987712	0.987808	0.988118	0.988215	0.988374	...	0.997785	0.998627	0.999259	0.999417	0.999327	0.9

29 rows × 29 columns

Las correlaciones se pueden ver en un mapa de color. Ya que es complejo, definamos la función “heatmap” para reutilizarla.

Input [3]:

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(rc={'figure.figsize':(12,6)})
pal=sns.diverging_palette(220, 20, n=100)
def heatmap(c):
  ax = sns.heatmap(c, vmin=-1, vmax=1, center=0, cmap=pal,square=True)
```

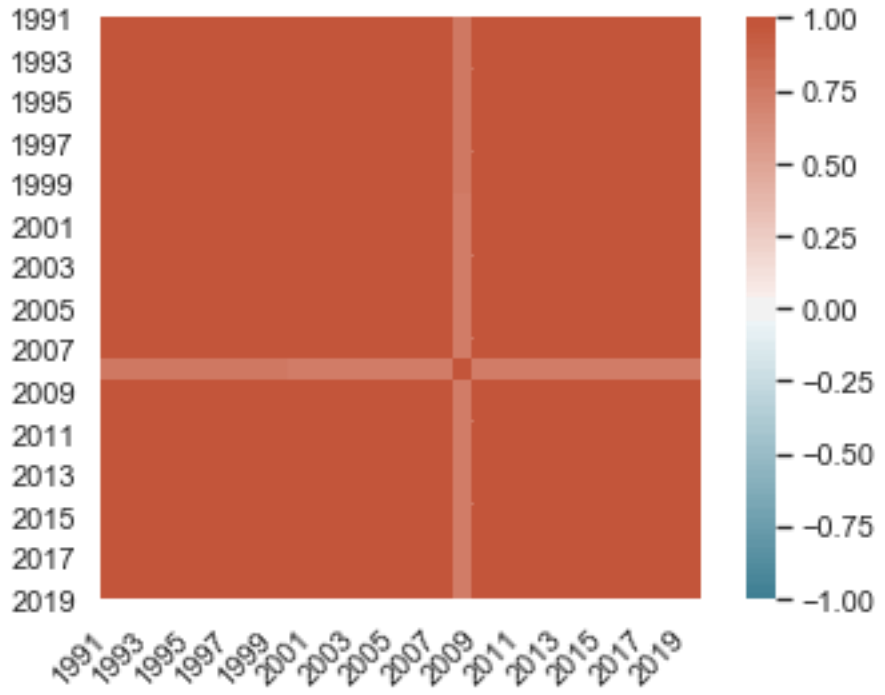
```
  ax.set_xticklabels(ax.get_xticklabels(), rotation=45,horizontalalignment='right');

sns.palplot(pal)
```

Output [3]:

Input [4]:

```
%matplotlib inline
corr = data.corr()
heatmap(corr)
```

Output [4]:



Puedes ver que los datos están altamente correlacionados anualmente

Los reorganizaremos para ver la correlación entre regiones transponiendo los datos.

- *dataframe.T transpone un conjunto de datos*

Input [5]:

```
dataT = data.T
dataT.head()
```

Output [5]:

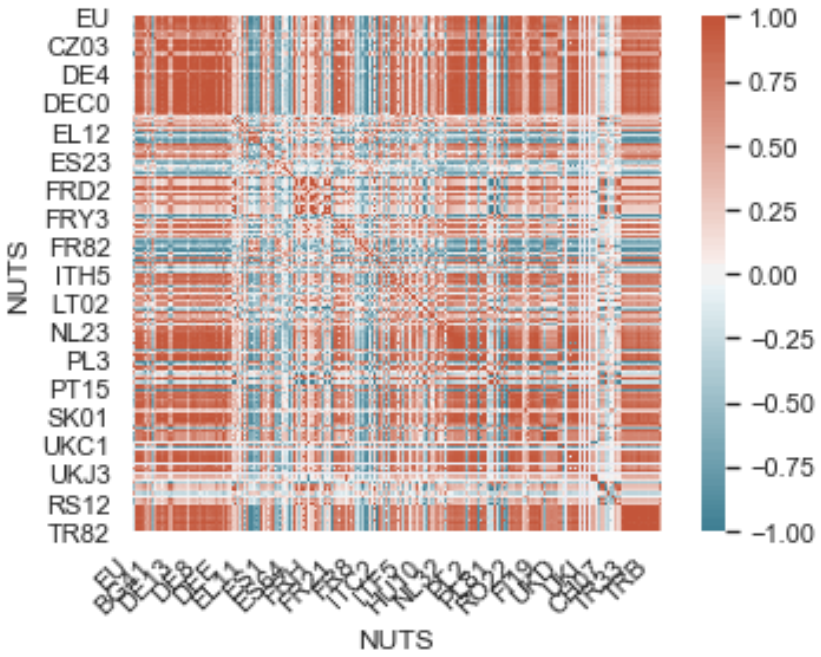
NUTS	EU	BE	BE1	BE10	BE2	BE21	BE22	BE23	BE24	BE25	...	TRA1	TRA2	TRB	TRB1	TRB2	TRC	TRC1	TRC2	TRC3	BA
1991	86785.24	3105.5	0.3	0.3	1655.2	357.4	183.5	440.1	141.9	532.4	...	263.9	292.7	290.7	117.2	173.5	187.9	36.2	107.4	44.3	462.0
1992	86785.24	3099.6	0.5	0.5	1661.4	361.4	180.8	439.9	139.4	540.0	...	263.9	292.7	290.7	117.2	173.5	187.9	36.2	107.4	44.3	462.0
1993	86785.24	3084.2	0.5	0.5	1637.2	362.9	179.6	419.5	144.9	530.3	...	263.9	292.7	290.7	117.2	173.5	187.9	36.2	107.4	44.3	462.0
1994	86785.24	3161.1	0.5	0.5	1685.5	378.3	183.4	442.2	145.6	536.1	...	263.9	292.7	290.7	117.2	173.5	187.9	36.2	107.4	44.3	462.0
1995	86785.24	3158.7	0.4	0.4	1678.5	368.5	182.9	439.8	148.3	539.0	...	263.9	292.7	290.7	117.2	173.5	187.9	36.2	107.4	44.3	462.0

5 rows × 537 columns

Input [6]:

```
corr = dataT.corr()
heatmap(corr)
```

Output [6]:



Se puede ver que en el caso de la producción anual entre regiones esta no está tan correlacionada, hay grupos de regiones que se comportan de forma similar pero otras no.

Para interpretar mejor los datos obtendremos una muestral de los datos centrada en España.

- *dataframe.iloc te permite elegir un rango de filas y columnas como un nuevo conjunto de datos.*

Input [7]:

```
dataTSub = dataT.loc[:,'ES':'ES70']
dataTSub.head()
```

Output [7]:

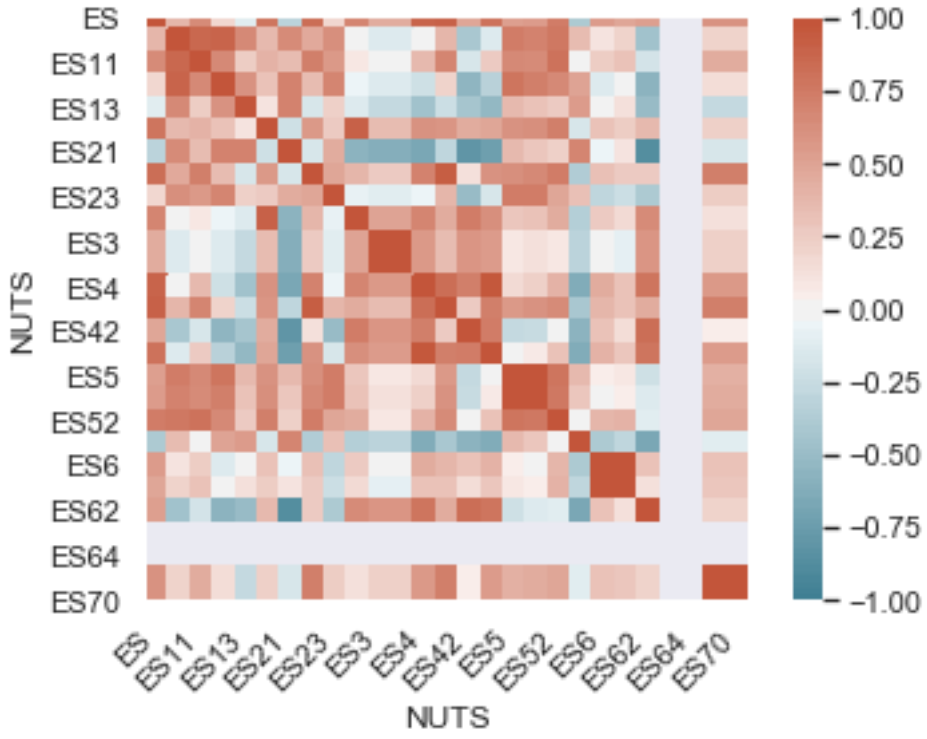
NUTS	ES	ES1	ES11	ES12	ES13	ES2	ES21	ES22	ES23	ES24	...	ES51	ES52	ES53	ES6	ES61	ES62	ES63	ES64	ES7	ES70
1991	5062.9	1621.6	868.3	403.1	350.2	497.6	180.3	94.0	38.8	184.5	...	558.9	36.7	51.0	509.4	467.9	41.4	0.0	0.0	16.7	16.7
1992	4975.6	1606.1	905.0	379.7	321.4	506.8	183.7	98.8	40.8	183.4	...	503.1	35.0	35.3	557.8	515.7	42.1	0.0	0.0	14.1	14.1
1993	5017.5	1591.6	884.7	402.5	304.4	518.7	181.0	95.9	42.2	199.6	...	511.3	37.8	31.3	597.1	556.9	40.3	0.0	0.0	16.8	16.8
1994	5251.0	1640.0	890.0	421.0	329.0	531.0	180.0	97.0	43.0	211.0	...	571.0	41.0	46.0	602.0	551.0	51.0	0.0	0.0	15.0	15.0
1995	5511.3	1690.5	953.9	409.5	327.1	578.5	187.8	99.7	50.4	240.6	...	647.1	52.5	54.9	551.7	522.7	29.0	0.0	0.0	17.9	17.9

5 rows × 27 columns

Input [8]:

```
corr = dataTSub.corr()
heatmap(corr)
```

Output [8]:



Si estamos interesados en saber si los datos observados son consistentes con una correlación debemos calcular el p-value.

Si el p-value está entre 0.0 y 0.05 se considera que hay correlación estadística evidente.

- *stats.pearsonr permite obtener el coeficiente de pearson y el p-value entre dos series de datos.*

Input [9]:

```
from scipy import stats

# Aragón (ES24) vs Cataluña (ES51)
pearson_coef, p_value = stats.pearsonr(dataTSub['ES24'], dataTSub['ES51'])
print("Pearson's correlation coefficient is {0:.8f} with a p-value of {1:.8f}".format(pearson_coef, p_value))

# Noreste (ES1) vs Galicia (ES11)
pearson_coef, p_value = stats.pearsonr(dataTSub['ES1'], dataTSub['ES11'])
print("Pearson's correlation coefficient is {0:.8f} with a p-value of {1:.8f}".format(pearson_coef, p_value))
```

Output [9]:

```
Pearson's correlation coefficient is 0.31009222 with a p-value of 0.10161273
Pearson's correlation coefficient is 0.86482739 with a p-value of 0.00000000
```

Visualización de las series de datos.

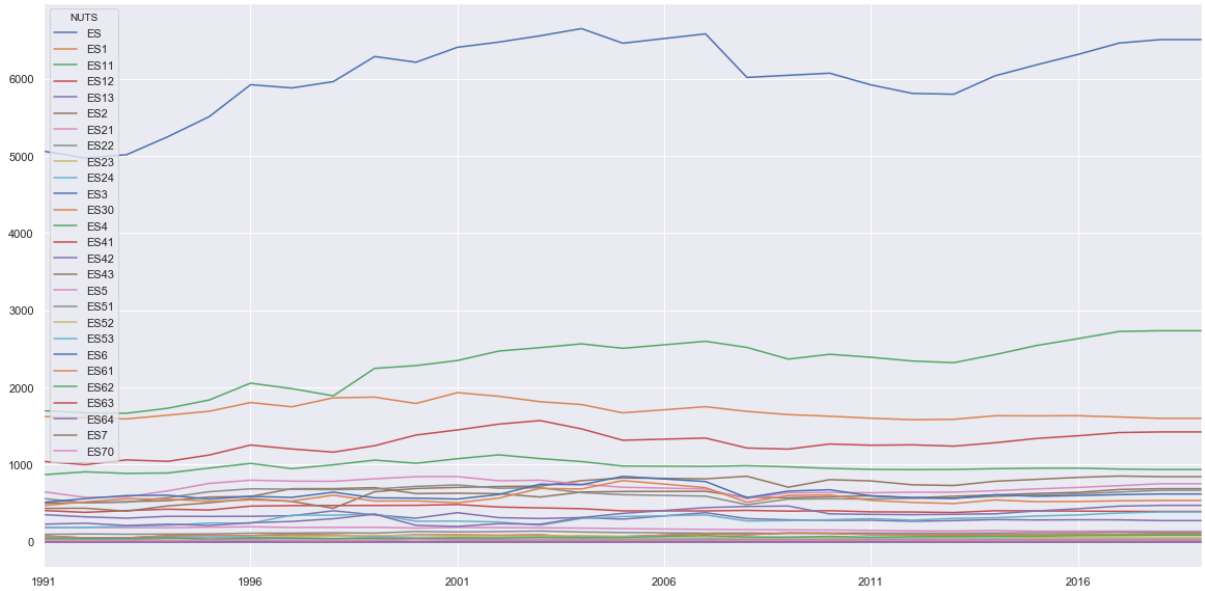
Una visión gráfica de los datos puede ayudarnos a entenderlos mejor.

- *dataframe.plot permite una amplia variedad de visualizaciones de diferentes tipos de datos.*

Input [10]

```
dataTSub.plot(kind='line', figsize =(20,10))

<matplotlib.axes._subplots.AxesSubplot at 0x26715d9f608>
```

Output [10]:



También puedes visualizar la correlación entre series usando un gráfico de relación linear.

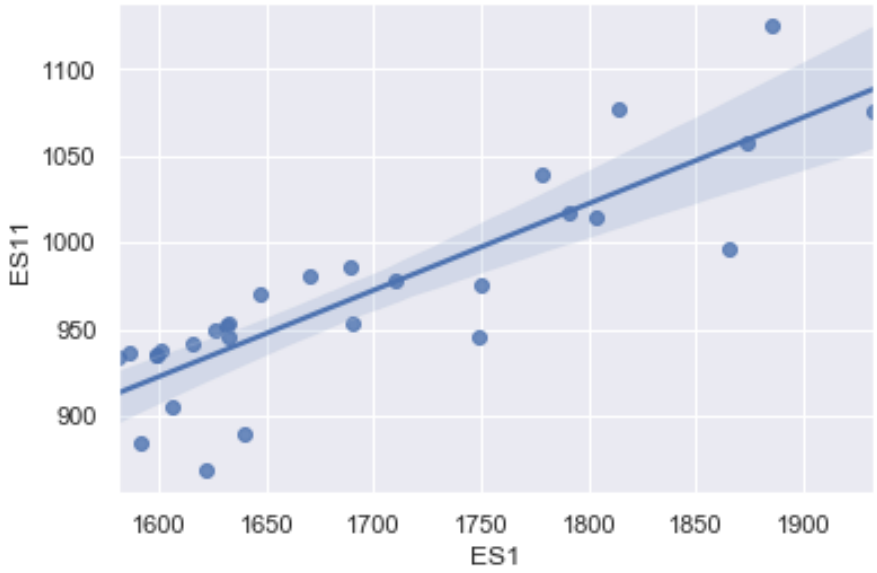
- *regplot() de la librería SeaBorn te permite mostra el grado de relación linear entre series*

Input [11]:

```
sns.regplot(x="ES1", y="ES11", data=dataTSub)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x26716a0b388>
```

Output [11]:

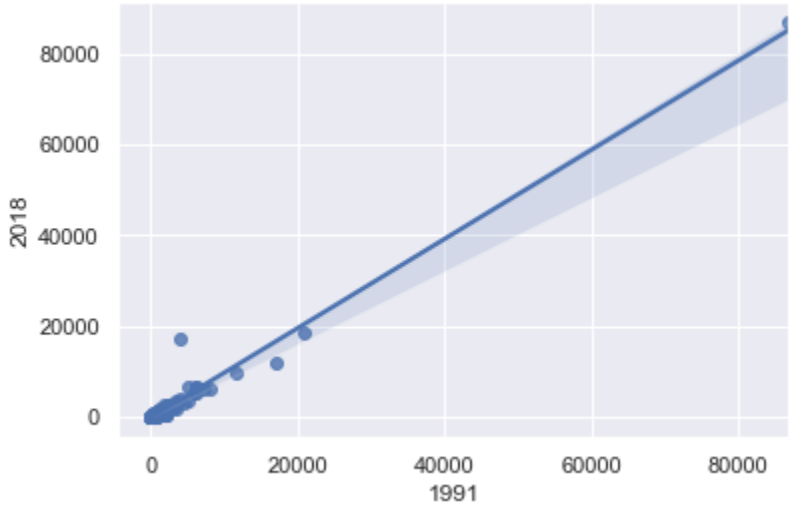


Input [12]:

```
sns.regplot(x="1991", y="2018", data=data)
```

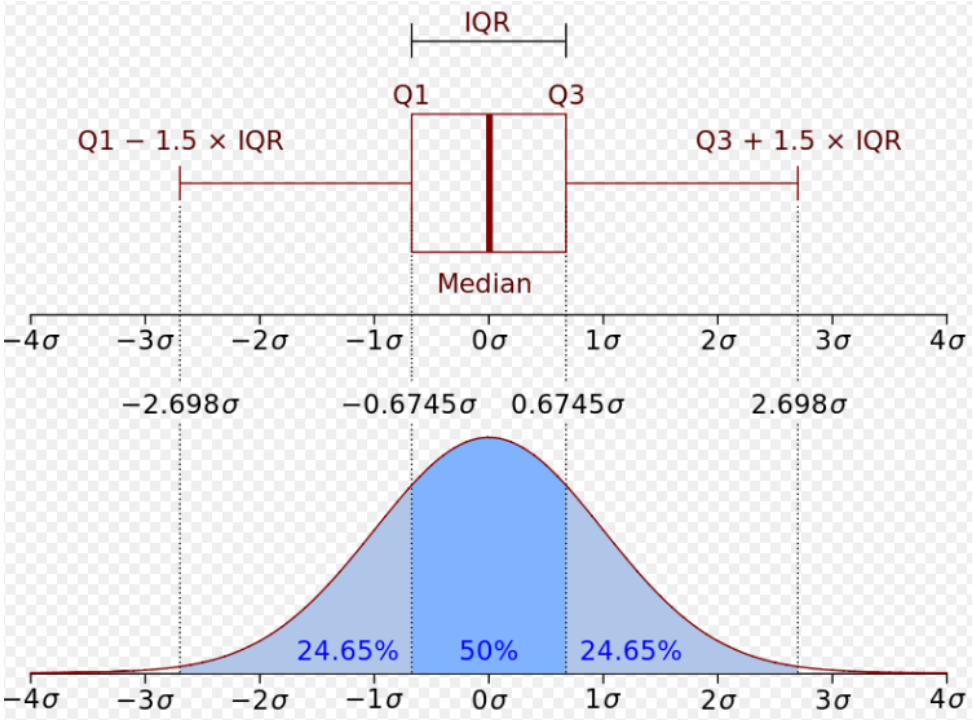
```
<matplotlib.axes._subplots.AxesSubplot at 0x26715b7c908>
```

Output [12]:



Si los datos a analizar son rangos, hay visualizaciones más informativas.

Es mejor usar un boxplot que muestra la dispersión de los valores respecto a las categorías.



El “boxplot” es una herramienta muy útil para identificar como una serie de datos se desvía de la media e incluso para ver si hay posibles datos espurios (muy alejados de la media).

- dataframe.boxplot te permite visualizar el boxplot de las columnas dese adas en la tabla.

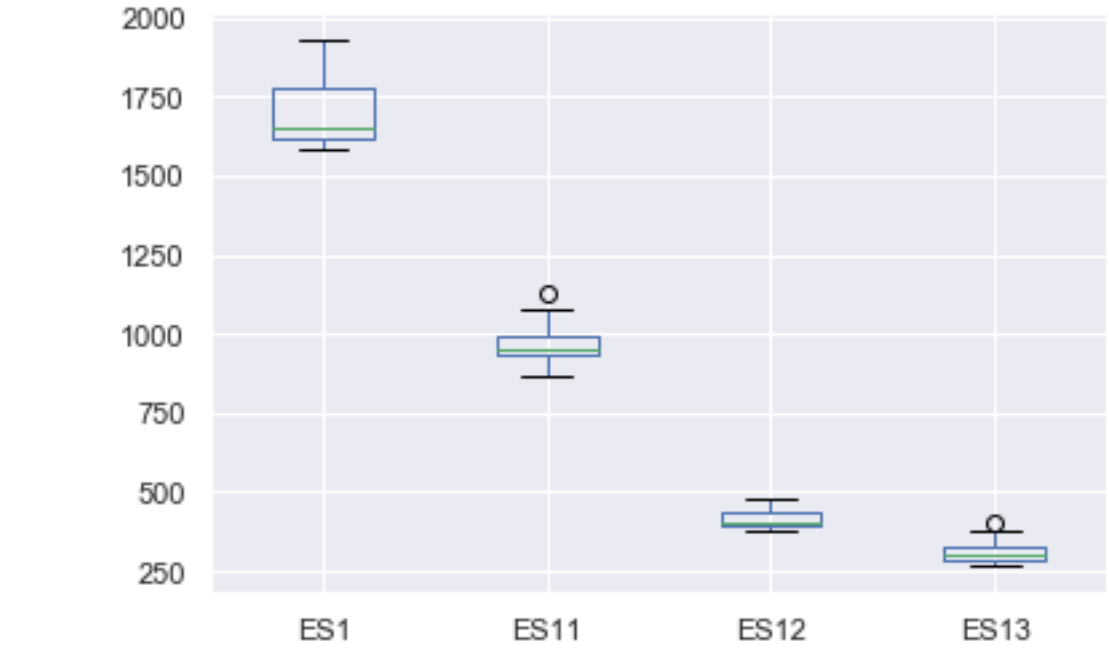
Input [13]:

```
import matplotlib.pyplot as plt
dataTSub[dataTSub.columns[1:5]].boxplot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x267167e8908>
```

Output [13]:

DATA ANALYSIS



Continua… Módulo 3 – Tarea 5