



APRENDIZAJE AUTOMÁTICO Y MINERÍA DE DATOS ACTIVIDAD GRUPAL

Análisis de datos

Instalar diversos paquetes de Python para cargar datos, conocer el
paquete Autoviz y analizar los resultados obtenidos

Profesor: Javier Martínez Torres

David Martínez del Valle
Miguel Soto Lombris
Ernesto González Pradas

ernesto.gonzalez023@comunidadunir.net

Índice

Contenido

Índice.....	1
Introducción.....	2
Diferenciación de los distintos tipos de datos	3
Calculo de los estadísticos básicos para una variable.....	4
Interpretar las gráficas: Histogramas, gráficas R y coeficiente de Pearson	7
Conclusión.....	11
Bibliografía	12
Hoja de control.....	13

Introducción

El objetivo de esta actividad es instalar y utilizar diversos paquetes de Python (Pandas, scikit-learn, matplotlib) para cargar ficheros de datos y realizar procesamiento de los mismos, instalar y conocer el paquete Autoviz para el Análisis Exploratorio de Datos (EDA) y analizar los resultados obtenidos por el EDA sobre un conjunto de datos real.

Para ello, se nos ha proporcionado con la actividad un fichero .csv (estaciones_madrid_202111.csv) en el que se encuentran los datos de las mediciones de distintos compuestos en el aire de Madrid por diferentes estaciones.

Diferenciación de los distintos tipos de datos

Lo primero que realizaremos en la actividad, será cargar los datos del csv suministrado (*estaciones_madrid_202111.csv*) para poder trabajar con los datos. Esto lo realizaremos con la librería Pandas tal y como se adjunta en la captura siguiente:

```
df=pd.read_csv('estaciones_madrid_202111.csv')
```

Con este comando ya tendríamos nuestros datos importados del csv y almacenado en un dataframe, el cual podemos mostrar de la siguiente forma:

```
2 df=pd.read_csv('estaciones_madrid_202111.csv')
df
```

	ANO	MES	DIA	MUNICIPIO	ESTACION	SO2	CO	NO	NO2	PM2.5	PM10	NOx	O3	TOL	BEN	EBE
0	2021	1	1	79	4	5.0	0.2	2.0	10.0	NaN	NaN	12.0	NaN	NaN	NaN	NaN
1	2021	1	1	79	8	2.0	0.2	4.0	23.0	3.0	4.0	29.0	51.0	0.6	0.4	0.2
2	2021	1	1	79	11	NaN	NaN	3.0	17.0	NaN	NaN	21.0	NaN	0.2	0.2	0.1
3	2021	1	1	79	16	NaN	NaN	3.0	18.0	NaN	NaN	22.0	51.0	NaN	NaN	NaN
4	2021	1	1	79	17	NaN	NaN	3.0	13.0	NaN	NaN	18.0	65.0	NaN	NaN	NaN
...
7913	2021	11	30	79	56	NaN	0.4	72.0	71.0	14.0	20.0	180.0	NaN	NaN	NaN	NaN
7914	2021	11	30	79	57	5.0	NaN	20.0	61.0	9.0	15.0	91.0	NaN	NaN	NaN	NaN
7915	2021	11	30	79	58	NaN	NaN	11.0	25.0	NaN	NaN	41.0	25.0	NaN	NaN	NaN
7916	2021	11	30	79	59	NaN	NaN	15.0	57.0	NaN	NaN	79.0	13.0	NaN	NaN	NaN
7917	2021	11	30	79	60	NaN	NaN	35.0	68.0	NaN	17.0	121.0	15.0	NaN	NaN	NaN

7918 rows × 16 columns

Como podemos observar, tenemos un dataframe con diferentes atributos (ANO, MES, DIA, MUNICIPIO...) y sus respectivos datos.

Lo primero que nos llama la atención son la cantidad de valores que tenemos del tipo NaN (Not a Number). En cuanto a la calidad del dato, podemos decir que estos son incompletos ya que presentan instancias de atributos que no toman ningún valor y decimos que su valor está ausente, “missing values”.

En cuanto al resto, podemos decir, que son atributos de tipo numéricos ya que toman valores numéricos reales, indicando una medida o cantidad.

Calculo de los estadísticos básicos para una variable

Siguiendo con lo visto en el apartado anterior, vamos a realizar unas series de transformaciones de dichos datos para poder estudiar mejor las gráficas que nos muestra el informe generado por Autoviz. Estas transformaciones se van a realizar variable a variable (de una en una).

La primera de las transformaciones es sustituir los valores NaN por 0 utilizando pandas:

```
104 df2 = df.fillna(0)
104 df2
```

	ANO	MES	DIA	MUNICIPIO	ESTACION	SO2	CO	NO	NO2	PM2.5	PM10	NOx	O3	TOL	BEN	EBE
0	2021	1	1	79	4	5.0	0.2	2.0	10.0	0.0	0.0	12.0	0.0	0.0	0.0	0.0
1	2021	1	1	79	8	2.0	0.2	4.0	23.0	3.0	4.0	29.0	51.0	0.6	0.4	0.2
2	2021	1	1	79	11	0.0	0.0	3.0	17.0	0.0	0.0	21.0	0.0	0.2	0.2	0.1
3	2021	1	1	79	16	0.0	0.0	3.0	18.0	0.0	0.0	22.0	51.0	0.0	0.0	0.0
4	2021	1	1	79	17	0.0	0.0	3.0	13.0	0.0	0.0	18.0	65.0	0.0	0.0	0.0
...
7913	2021	11	30	79	56	0.0	0.4	72.0	71.0	14.0	20.0	180.0	0.0	0.0	0.0	0.0
7914	2021	11	30	79	57	5.0	0.0	20.0	61.0	9.0	15.0	91.0	0.0	0.0	0.0	0.0
7915	2021	11	30	79	58	0.0	0.0	11.0	25.0	0.0	0.0	41.0	25.0	0.0	0.0	0.0
7916	2021	11	30	79	59	0.0	0.0	15.0	57.0	0.0	0.0	79.0	13.0	0.0	0.0	0.0
7917	2021	11	30	79	60	0.0	0.0	35.0	68.0	0.0	17.0	121.0	15.0	0.0	0.0	0.0

7918 rows x 16 columns

En esta captura observamos como todos los datos que aparecían como NaN ahora están sustituidos por 0. Podríamos utilizar también la función `.dropna()` pero nos dimos cuenta que para determinados atributos en algunos periodos de tiempo nos quedábamos sin datos para poder realizar el estudio posterior con Autoviz. Este es el caso del atributo PM2.5 por ejemplo.

A continuación, extraemos un dataframe de todos los registros filtrando por la estación 4 (podríamos filtrar a la vez por mes y días):

```
7 dfEstacion4 = df2[df2['ESTACION'] == 4]
dfEstacion4
```

	ANO	MES	DIA	MUNICIPIO	ESTACION	SO2	CO	NO	NO2	PM2.5	PM10	NOx	O3	TOL	BEN	EBE
0	2021	1	1	79	4	5.0	0.2	2.0	10.0	0.0	0.0	12.0	0.0	0.0	0.0	0.0
24	2021	1	2	79	4	5.0	0.3	5.0	23.0	0.0	0.0	30.0	0.0	0.0	0.0	0.0
48	2021	1	3	79	4	5.0	0.4	10.0	29.0	0.0	0.0	45.0	0.0	0.0	0.0	0.0
72	2021	1	4	79	4	5.0	0.3	9.0	29.0	0.0	0.0	43.0	0.0	0.0	0.0	0.0
96	2021	1	5	79	4	6.0	0.5	49.0	54.0	0.0	0.0	129.0	0.0	0.0	0.0	0.0
...
6950	2021	10	20	79	4	4.0	0.4	34.0	40.0	0.0	0.0	92.0	0.0	0.0	0.0	0.0
6974	2021	10	21	79	4	3.0	0.4	20.0	36.0	0.0	0.0	66.0	0.0	0.0	0.0	0.0
6998	2021	10	22	79	4	3.0	0.2	6.0	24.0	0.0	0.0	33.0	0.0	0.0	0.0	0.0
7022	2021	10	23	79	4	4.0	0.3	10.0	36.0	0.0	0.0	52.0	0.0	0.0	0.0	0.0
7046	2021	10	24	79	4	4.0	0.3	16.0	42.0	0.0	0.0	66.0	0.0	0.0	0.0	0.0

266 rows × 16 columns

Una segunda forma de transformar los datos, esta vez utilizando la librería scikit-learn, sería sustituir los NaN por el valor promedio de los datos legibles en la columna del atributo correspondiente. De forma alternativa se podrían a ver sustituido por la mediana de cada columna en función del valor que se le pase al atributo strategy del método *SimpleImputer()*. El método devuelve un array al cual hay que volver a dar forma de DataFrame para poder realizar una correcta visualización de los datos.

```
9 imp_mean = SimpleImputer(missing_values=np.nan, strategy='mean')
imp_mean.fit(df)
columns=['ANO','MES','DIA','MUNICIPIO','ESTACION','SO2','CO','NO','NO2','PM2.5','PM10','NOx','O3','TOL','BEN','EBE']
df3 = round(pd.DataFrame(imp_mean.transform(df), columns=columns),1)
df3
```

	ANO	MES	DIA	MUNICIPIO	ESTACION	SO2	CO	NO	NO2	PM2.5	PM10	NOx	O3	TOL	BEN	EBE
0	2021.0	1.0	1.0	79.0	4.0	5.0	0.2	2.0	10.0	9.8	18.9	12.0	53.6	1.4	0.3	0.3
1	2021.0	1.0	1.0	79.0	8.0	2.0	0.2	4.0	23.0	3.0	4.0	29.0	51.0	0.6	0.4	0.2
2	2021.0	1.0	1.0	79.0	11.0	5.1	0.3	3.0	17.0	9.8	18.9	21.0	53.6	0.2	0.2	0.1
3	2021.0	1.0	1.0	79.0	16.0	5.1	0.3	3.0	18.0	9.8	18.9	22.0	51.0	1.4	0.3	0.3
4	2021.0	1.0	1.0	79.0	17.0	5.1	0.3	3.0	13.0	9.8	18.9	18.0	65.0	1.4	0.3	0.3
...
7913	2021.0	11.0	30.0	79.0	56.0	5.1	0.4	72.0	71.0	14.0	20.0	180.0	53.6	1.4	0.3	0.3
7914	2021.0	11.0	30.0	79.0	57.0	5.0	0.3	20.0	61.0	9.0	15.0	91.0	53.6	1.4	0.3	0.3
7915	2021.0	11.0	30.0	79.0	58.0	5.1	0.3	11.0	25.0	9.8	18.9	41.0	25.0	1.4	0.3	0.3
7916	2021.0	11.0	30.0	79.0	59.0	5.1	0.3	15.0	57.0	9.8	18.9	79.0	13.0	1.4	0.3	0.3
7917	2021.0	11.0	30.0	79.0	60.0	5.1	0.3	35.0	68.0	9.8	17.0	121.0	15.0	1.4	0.3	0.3

7918 rows × 16 columns

Una vez obtenidos estos datos utilizando la librería scikit-learn, pasamos a normalizar el fichero utilizando la media:

185

df_aux = (df2.iloc[:,0:5:])

df_aux_mitad2 = pd.DataFrame(df2.iloc[:,5:])

for column in df_aux_mitad2:

df_aux_mitad2[column] = (df_aux_mitad2[column] - df_aux_mitad2[column].mean()) / (df_aux_mitad2[column].std())

df_normalizado = pd.concat([df_aux, df_aux_mitad2], axis=1)

df_normalizado

185

	ANO	MES	DIA	MUNICIPIO	ESTACION	SO2	CO	NO	NO2	PM2.5	PM10	NOx	O3	TOL	BEN	EBE
0	2021	1	1	79	4	1.666065	1.072560	-0.439704	-1.060436	-0.528647	-0.725602	-0.753406	-0.938081	-0.394776	-0.415831	-0.340741
1	2021	1	1	79	8	0.419036	1.072560	-0.311537	-0.294168	-0.031183	-0.435610	-0.320482	0.718726	0.315749	1.982675	0.632208
2	2021	1	1	79	11	-0.412316	-0.455148	-0.375621	-0.647830	-0.528647	-0.725602	-0.524211	-0.938081	-0.157934	0.783422	0.145733
3	2021	1	1	79	16	-0.412316	-0.455148	-0.375621	-0.588886	-0.528647	-0.725602	-0.498745	0.718726	-0.394776	-0.415831	-0.340741
4	2021	1	1	79	17	-0.412316	-0.455148	-0.375621	-0.883605	-0.528647	-0.725602	-0.600609	1.173536	-0.394776	-0.415831	-0.340741
...
7913	2021	11	30	79	56	-0.412316	2.600269	4.046119	2.535130	1.792853	0.724357	3.524907	-0.938081	-0.394776	-0.415831	-0.340741
7914	2021	11	30	79	57	1.666065	-0.455148	0.713793	1.945693	0.963745	0.361867	1.258420	-0.938081	-0.394776	-0.415831	-0.340741
7915	2021	11	30	79	58	-0.412316	-0.455148	0.137045	-0.176280	-0.528647	-0.725602	-0.014888	-0.125921	-0.394776	-0.415831	-0.340741
7916	2021	11	30	79	59	-0.412316	-0.455148	0.393377	1.709918	-0.528647	-0.725602	0.952826	-0.515758	-0.394776	-0.415831	-0.340741
7917	2021	11	30	79	60	-0.412316	-0.455148	1.675041	2.358299	-0.528647	0.506863	2.022404	-0.450785	-0.394776	-0.415831	-0.340741

Otra forma de normalizar el dataFrame obtenido anteriormente, es utilizando el método min-max:

187

df_aux = (df2.iloc[:,0:5:])

df_aux_mitad2 = pd.DataFrame(df2.iloc[:,5:])

for column in df_aux_mitad2:

df_aux_mitad2[column] = (df_aux_mitad2[column] - df_aux_mitad2[column].min()) / (df_aux_mitad2[column].max() - df_aux_mitad2[column].min())

df_normalizado_min_max = pd.concat([df_aux, df_aux_mitad2], axis=1)

df_normalizado_min_max

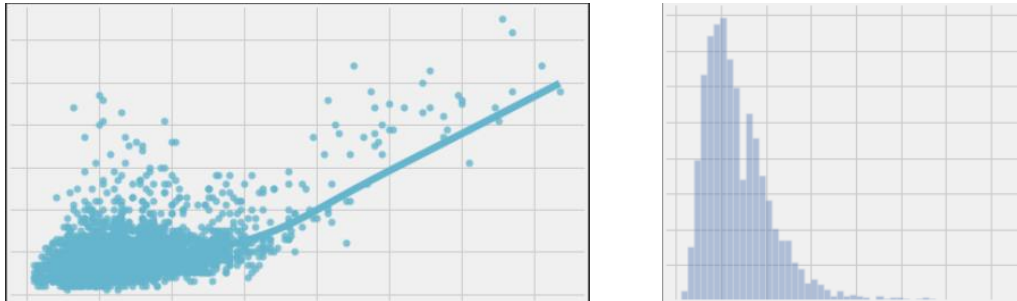
187

	ANO	MES	DIA	MUNICIPIO	ESTACION	SO2	CO	NO	NO2	PM2.5	PM10	NOx	O3	TOL	BEN	EBE
0	2021	1	1	79	4	0.3125	0.181818	0.009346	0.068027	0.000000	0.000	0.025974	0.000000	0.000000	0.000000	0.000000
1	2021	1	1	79	8	0.1250	0.181818	0.018692	0.156463	0.046154	0.032	0.062771	0.463636	0.040268	0.173913	0.030303
2	2021	1	1	79	11	0.0000	0.000000	0.014019	0.115646	0.000000	0.000	0.045455	0.000000	0.013423	0.086957	0.015152
3	2021	1	1	79	16	0.0000	0.000000	0.014019	0.122449	0.000000	0.000	0.047619	0.463636	0.000000	0.000000	0.000000
4	2021	1	1	79	17	0.0000	0.000000	0.014019	0.088435	0.000000	0.000	0.038961	0.590909	0.000000	0.000000	0.000000
...
7913	2021	11	30	79	56	0.0000	0.363636	0.336449	0.482993	0.215385	0.160	0.389610	0.000000	0.000000	0.000000	0.000000
7914	2021	11	30	79	57	0.3125	0.000000	0.093458	0.414966	0.138462	0.120	0.196970	0.000000	0.000000	0.000000	0.000000
7915	2021	11	30	79	58	0.0000	0.000000	0.051402	0.170068	0.000000	0.000	0.088745	0.227273	0.000000	0.000000	0.000000
7916	2021	11	30	79	59	0.0000	0.000000	0.070093	0.387755	0.000000	0.000	0.170996	0.118182	0.000000	0.000000	0.000000
7917	2021	11	30	79	60	0.0000	0.000000	0.163551	0.462585	0.000000	0.136	0.261905	0.136364	0.000000	0.000000	0.000000

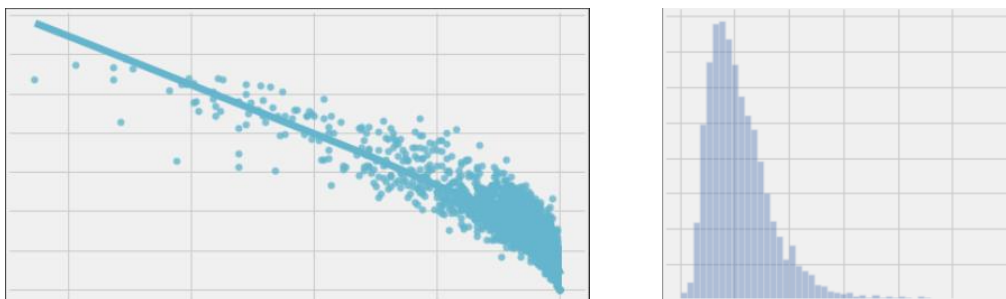
7918 rows x 16 columns

Interpretar las gráficas: Histogramas, gráficas R y coeficiente de Pearson

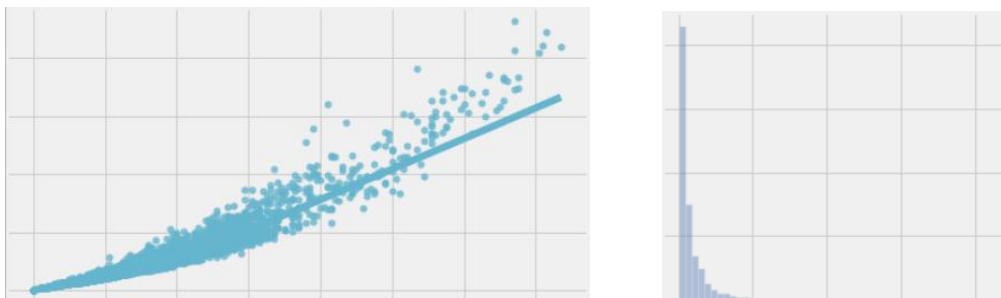
Lo primero vamos a ver las gráficas generadas con Autoviz en las distintas situaciones vistas con anterioridad. Primero observamos la gráfica del dataframe con los valores NaN:



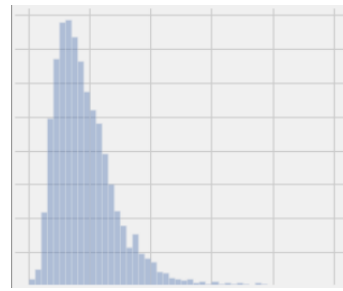
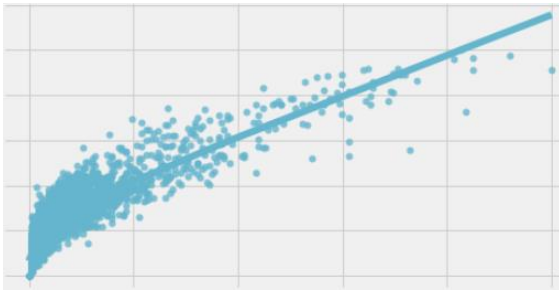
Ahora pasamos a observar las gráficas una vez que sustituimos los Nan por 0 en pandas:



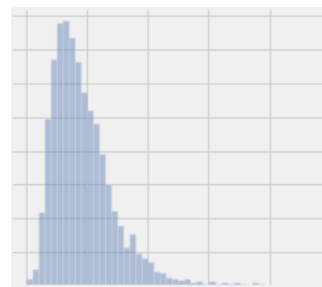
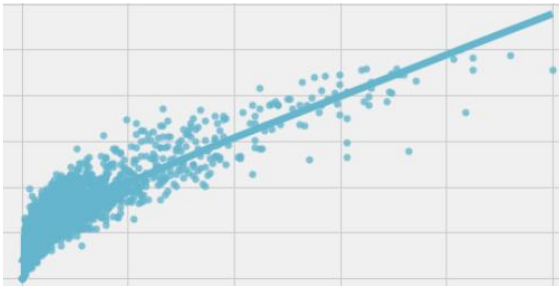
Las siguientes gráficas muestran los datos del dataframe que estaban a Nan sustituidos por el promedio, realizado con la librería scikit-learn:



Normalizando el fichero utilizando la media, estas serían sus gráficas:



Y por último, normalizando el fichero utilizando el método de min-max:



A continuación, vamos a comprobar si las variables de SO2 y CO de una estación concreta, en un mes determinado están relacionadas. Para ello vamos a calcular el coeficiente de correlación de Pearson, cuya fórmula sería la siguiente:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

El coeficiente de correlación de Pearson oscila entre -1 y +1. A continuación, mostramos las capturas de pantalla del código con Python y los resultados y posteriormente comentaremos el resultado obtenido. Este sería el código utilizado:

```
PracticaGrupal.py
1 import pandas as pd
2 import math
3
4 #Cargamos los datos del csv y los metemos en un dataframe
5 df = pd.read_csv('estaciones_madrid_202111.csv', sep=",")
6
7 #Extraemos en un dataframe todos los registros de la estación 4 (por ejemplo)
8 dfEstacion4 = df[df['ESTACION'] == 4]
9 #Extraemos en un dataframe todos los registros de la estación 4 en el mes de Junio (6)
10 dfEstacion4IntervaloTiempo = dfEstacion4[dfEstacion4['MES'] == 6]
11 #Extraemos en un dataframe todos los registros seleccionados solo con las columnas que nos interesan
12 df2 = dfEstacion4IntervaloTiempo[['ESTACION', 'MES', 'DIA', 'SO2', 'CO']]
```

```

14 #Realizamos los cálculos necesarios para posteriormente introducirlos en la fórmula de correlacion de Pearson
15 df2['(S02*C0)'] = df2['S02'] * df2['C0']
16 df2['S02^2'] = df2['S02'] * df2['S02']
17 df2['C0^2'] = df2['C0'] * df2['C0']
18 print(df2)
19
20
21 totalS02 = df2['S02'].sum()
22 print("El total de la Columna S02 es: ", totalS02)
23 totalC0 = df2['C0'].sum()
24 print("El total de la Columna C0 es: ", totalC0)
25 totalS02porC0 = df2['(S02*C0)'].sum()
26 print("El total de la columna S02*C0 es: ", totalS02porC0)
27 totalS02Cuadrado = df2['S02^2'].sum()
28 print("El total de la Columna S02 al cuadrado es: ", totalS02Cuadrado)
29 totalC0Cuadrado = df2['C0^2'].sum()
30 print("El total de la Columna C0 al cuadrado es: ", totalC0Cuadrado)

```

```

#Calculamos el coeficiente de correlación de Pearson
#n es el número de registros, como es el mes de Junio son 30 días
n = 30
#Fórmula matemática para calcular la correlación de Pearson. Introducimos todos los datos calculados
correlacionPearson = ((n * totalS02porC0) - (totalS02 * totalC0)) / (math.sqrt(n * totalS02Cuadrado - (totalS02 * totalS02))) * (math.sqrt(n * totalC0Cuadrado - (totalC0 * totalC0)))
print("La correlación de Pearson entre S02 y C0 es: ", correlacionPearson)

```

Y el resultado sería el siguiente:

ESTACION	MES	DIA	S02	C0	(S02*C0)	S02^2	C0^2
3600	4	6	1	10.0	0.2	2.0	100.0
3624	4	6	2	10.0	0.2	2.0	100.0
3648	4	6	3	10.0	0.2	2.0	100.0
3672	4	6	4	10.0	0.2	2.0	100.0
3696	4	6	5	11.0	0.2	2.2	121.0
3720	4	6	6	10.0	0.2	2.0	100.0
3743	4	6	7	10.0	0.2	2.0	100.0
3766	4	6	8	9.0	0.2	1.8	81.0
3789	4	6	9	9.0	0.2	1.8	81.0
3812	4	6	10	9.0	0.2	1.8	81.0
3835	4	6	11	9.0	0.2	1.8	81.0
3859	4	6	12	9.0	0.2	1.8	81.0
3883	4	6	13	9.0	0.2	1.8	81.0
3907	4	6	14	9.0	0.2	1.8	81.0
3931	4	6	15	9.0	0.2	1.8	81.0
3955	4	6	16	10.0	0.2	2.0	100.0
3979	4	6	17	11.0	0.2	2.2	121.0
4003	4	6	18	11.0	0.2	2.2	121.0
4027	4	6	19	10.0	0.2	2.0	100.0
4051	4	6	20	11.0	0.1	1.1	121.0
4075	4	6	21	11.0	0.2	2.2	121.0
4099	4	6	22	11.0	0.2	2.2	121.0
4123	4	6	23	11.0	0.2	2.2	121.0
4146	4	6	24	10.0	0.2	2.0	100.0
4169	4	6	25	10.0	0.2	2.0	100.0
4192	4	6	26	9.0	0.2	1.8	81.0
4215	4	6	27	9.0	0.2	1.8	81.0
4238	4	6	28	9.0	0.2	1.8	81.0
4261	4	6	29	9.0	0.2	1.8	81.0
4284	4	6	30	10.0	0.2	2.0	100.0
El total de la Columna S02 es: 295.0							
El total de la Columna C0 es: 5.900000000000002							
El total de la columna S02*C0 es: 57.89999999999999							
El total de la Columna S02 al cuadrado es: 2919.0							
El total de la Columna C0 al cuadrado es: 1.1700000000000004							
La correlación de Pearson entre S02 y C0 es: -0.08073629062482533							

Como podemos observar, el resultado que nos sale es: -0.0807. Como hemos dicho anteriormente, el coeficiente de correlación de Pearson varía entre -1 y +1. En este caso al ser un valor menor que 0 indica que existe una correlación negativa, es decir, que las variables seleccionadas (SO₂ y CO) están asociadas en sentido inverso. Cuánto más se acerque esta correlación a -1, mayor es la relación invertida. A continuación, adjuntamos una tabla para interpretar el coeficiente de correlación en forma de escala:

Valor	Significado
-1	Correlación negativa grande y perfecta
-0,9 a -0,99	Correlación negativa muy alta
-0,7 a -0,89	Correlación negativa alta
-0,4 a -0,69	Correlación negativa moderada
-0,2 a -0,39	Correlación negativa baja
-0,01 a -0,19	Correlación negativa muy baja
0	Correlación nula
0,01 a 0,19	Correlación positiva muy baja
0,2 a 0,39	Correlación positiva baja
0,4 a 0,69	Correlación positiva moderada
0,7 a 0,89	Correlación positiva alta
0,9 a 0,99	Correlación positiva muy alta
1	Correlación positiva grande y perfecta

Conclusión

Con esta actividad grupal, hemos podido practicar y ver gráficamente como cargar datos importados desde un csv y operar con las distintas columnas, normalizando los datos en base a distintas operaciones y opciones que se nos plantean a lo largo del temario.

Hemos podido manejar diferentes librerías como pandas para trabajar con dataframe, scikit-learn para normalización de datos, matplotlib para representación de gráficas y Autoviz para mostrar el informe automático que genera con sus respectivas gráficas.

Pandas es una tecnología muy potente, que nos permite trabajar con grandes cantidades de datos, pudiendo manejar dichos datos de muy diversas formas, seleccionando columnas, operando con ellas o con valores concretos de dichas columnas para posteriormente introducir los resultados en una fórmula como la del coeficiente de correlación de Pearson e interpretar los resultados obtenidos.

Bibliografía

Chacón, J. L. (22 de 03 de 2021). *profile.es*. Obtenido de <https://profile.es/blog/pandas-python/>

GeeksforGeeks. (24 de Enero de 2021). *GeeksforGeeks*. Obtenido de <https://www.geeksforgeeks.org/how-to-fill-nan-values-with-mean-in-pandas/>

pandas.pydata.org. (2022). *pandas.pydata.org*. Obtenido de https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html

ROVIRA, A. C. (2022). *Tema 3. Exploración y preprocesamiento*.

scikit-learn. (Diciembre de 2019). *scikit-learn*. Obtenido de <https://scikit-learn.org/stable/>

Hoja de control

HOJA DE CONTROL ACTIVIDAD GRUPAL 1			
Nombre y apellidos del primer miembro del equipo: David			
Asistencia a reuniones de equipo por cada integrante (se incluirá una línea por cada miembro del equipo recogiendo su nombre)	Marcar con una X lo que proceda		
	Asistencia a una reunión o ninguna <input type="checkbox"/>	Asistencia a dos reuniones <input checked="" type="checkbox"/>	Asistencia a tres reuniones <input type="checkbox"/>
Tareas o entregas a realizadas por cada integrante (se incluirá una línea por cada miembro del equipo recogiendo su nombre)	Ninguna o una tarea <input type="checkbox"/>	Dos tareas <input checked="" type="checkbox"/>	Tres tareas <input type="checkbox"/>
OBSERVACIONES			

HOJA DE CONTROL ACTIVIDAD GRUPAL 1			
Nombre y apellidos del primer miembro del equipo: Miguel			
Asistencia a reuniones de equipo por cada integrante (se incluirá una línea por cada miembro del equipo recogiendo su nombre)	Marcar con una X lo que proceda		
	Asistencia a una reunión o ninguna <input type="checkbox"/>	Asistencia a dos reuniones <input checked="" type="checkbox"/>	Asistencia a tres reuniones <input type="checkbox"/>
Tareas o entregas a realizadas por cada integrante (se incluirá una línea por cada miembro del equipo recogiendo su nombre)	Ninguna o una tarea <input type="checkbox"/>	Dos tareas <input checked="" type="checkbox"/>	Tres tareas <input type="checkbox"/>
OBSERVACIONES			

HOJA DE CONTROL ACTIVIDAD GRUPAL 1			
Nombre y apellidos del primer miembro del equipo: Ernesto			
Asistencia a reuniones de equipo por cada integrante (se incluirá una línea por cada miembro del equipo recogiendo su nombre)	Marcar con una X lo que proceda		
	Asistencia a una reunión o ninguna <input type="checkbox"/>	Asistencia a dos reuniones <input checked="" type="checkbox"/>	Asistencia a tres reuniones <input type="checkbox"/>

Tareas o entregas a realizadas por cada integrante (se incluirá una línea por cada miembro del equipo recogiendo su nombre)	Ninguna o una tarea <input type="checkbox"/>	Dos tareas <input checked="" type="checkbox"/>	Tres tareas <input type="checkbox"/>
OBSERVACIONES			