Jordi Bosch Alibau

Práctica 2

ENUNCIADO

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

Para hacer esta práctica tendréis que trabajar en grupos de 2 personas. Tendréis que entregar un solo archivo con el enlace Github (https://github.com) donde se encuentren las soluciones incluyendo los nombres de los componentes del equipo.

Podéis utilizar la Wiki de Github para describir vuestro equipo y los diferentes archivos que corresponden a vuestra entrega. Cada miembro del equipo tendrá que contribuir con su usuario Github.

Aunque no se trata del mismo enunciado, los siguientes ejemplos de ediciones anteriores os pueden servir como guía:

- ***Ejemplo: https://github.com/Bengis/nba-gap-cleaning***
- ***Ejemplo complejo (archivo Campus)***

El objetivo de esta actividad será el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien cualquier dataset libre disponible en Kaggle (https://www.kaggle.com).

Algunos ejemplos de dataset con los que podéis trabajar son:

- ***Red Wine Quality (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009)***
- ***Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic)***

El último ejemplo corresponde a una competición activa de Kaggle de manera que, opcionalmente, podéis aprovechar el trabajo realizado durante la práctica para entrar en estacompetición.

Los siguientes recursos son de utilidad para la realización de la práctica:

- ***Calvo M., Subirats L., Pérez D. (2019). Introducción a la limpieza y análisis de los datos. Ed. UOC.***
- ***M. Squire (2015). Clean Data. Packt Publishing Ltd.***
- ***J. Han, M. Kamber, J. Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.***
- ***J. W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369 .***
- ***P. Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.***
- ***W. McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.***
- ***Tutorial de Github https://guides.github.com/activities/hello-world.***

RESPUESTA

ENLACE GITHUB > https://github.com/jordiba90/jordiba90_prac2

1) Descripción del dataset

```
##############################################################################
#Inspect a DataFrame - Shape and Size > Obtendremos el tamaño y el número de registros/variables
##############################################################################
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
print("\n\nNúmero de registros y variables del dataset:",tcd_df.shape)
print("Número de registros del dataset:",tcd_df.shape[0])
print("Número de variables del dataset:",tcd_df.shape[1])
print("Número de campos del dataset:",tcd_df.size)
```

```
Número de registros y variables del dataset: (223, 23)
Número de registros del dataset: 223
Número de variables del dataset: 23
Número de campos del dataset: 5129
```

```
##############################################################################
#Columns > Obtendremos datos acerca del nombre de las variables
##############################################################################
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
print("\nLista de variables del dataset:\n")
print(tcd_df.columns)
```

```
Lista de variables del dataset:

Index(['album_uri', 'album_name', 'album_img', 'album_release_date',
       'album_release_year', 'album_popularity', 'track_name', 'track_uri',
       'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness',
       'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo',
       'duration_ms', 'time_signature', 'key_mode', 'track_popularity'],
      dtype='object')
```

RESPUESTA

1) Descripción del dataset

```
#############################################################################
#Inspect a DataFrame - Info > Obtendremos datos sobre las variables de estudio
#############################################################################
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
print("Información sobre el dataset:\n")
print(tcd_df.info())
```

```
Información sobre el dataset:

<class 'pandas.core.frame.DataFrame'>
Int64Index: 223 entries, 1 to 240
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   album_uri          223 non-null    object
 1   album_name         223 non-null    object
 2   album_img          223 non-null    object
 3   album_release_date 223 non-null    object
 4   album_release_year 223 non-null    object
 5   album_popularity   223 non-null    int64
 6   track_name         223 non-null    object
 7   track_uri          223 non-null    object
 8   danceability       223 non-null    float64
 9   energy             223 non-null    float64
 10  key                223 non-null    object
 11  loudness           223 non-null    float64
 12  mode               223 non-null    object
 13  speechiness        223 non-null    float64
 14  acousticness       223 non-null    float64
 15  instrumentalness   223 non-null    float64
 16  liveness           223 non-null    float64
 17  valence            223 non-null    float64
 18  tempo              223 non-null    float64
 19  duration_ms        223 non-null    int64
 20  time_signature     223 non-null    int64
 21  key_mode           223 non-null    object
 22  track_popularity   223 non-null    int64
dtypes: float64(9), int64(4), object(10)
memory usage: 41.8+ KB
None
```

1.1) Por qué es importante?

Se trata de mi grupo favorito desde la adolescencia.

1.2) Qué pregunta pretende responder?

Quisiera averiguar la correlación entre las variables “energy” y “loudness” para comprobar si una mayor energía en la voz del cantante influye en el volumen de las canciones a lo largo del conjunto de la discografía que conforma el dataset.

RESPUESTA

2) Integración y selección de los datos

2.1) Integración

```
#########################################################################
#Aggregation
#########################################################################
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
print("\nAgrupación de datos según la variable \"album_name\" filtrando la variable \"duration_ms\" para calcular el mínimo, la media, y el máximo de la duración en minutos de las canciones:\n")
print(tcd_df.groupby('album_name')['duration_ms'].agg(['min', np.mean, max])/1000/60)
```

```
Agrupación de datos según la variable "album_name" filtrando la variable "duration_ms"

                                                    min      mean        max
album_name
4:13 Dream                                     2.374000  4.054035   6.276450
Bestival Live 2011                             2.979717  4.865672   8.526917
Bloodflowers                                   3.714883  6.435580  11.204883
Concert - The Cure Live                        2.774783  4.250590   6.813517
Disintegration (Deluxe Edition [Remastered])   2.686450  4.824512   9.381783
Hypnagogic States                              3.265333  6.903261  21.443333
Kiss Me Kiss Me Kiss Me                        2.380217  4.068088   6.964450
Mixed Up (Remastered 2018 / Deluxe Edition)    3.721783  6.638488   8.806000
Paris                                          2.690000  4.802775   7.448883
Pornography                                    4.374883  5.415971   6.683333
Show                                           2.464450  4.929674   7.975550
The Cure                                       2.962000  4.678162  10.275333
The Head On The Door                           2.612500  3.767970   4.837483
The Top                                        2.341117  3.812100   6.967333
Wild Mood Swings                               2.647783  4.410619   7.937783
Wish                                           3.555550  5.482983   7.667783
```

2.2) Selección

Seleccionaremos las variables “energy” y “loudness”.

RESPUESTA

3) Limpieza de los datos

3.1) Los datos contienen ceros o elementos vacíos?

No, he tenido la suerte de que el dataset no contenía valores nulos pues al comprobar el número de 223 registros multiplicados por las 23 variables obtenemos el total de 5.129 campos que forman tal como puede observarse en la descripción del dataset:

```
Número de registros y variables del dataset: (223, 23)
Número de registros del dataset: 223
Número de variables del dataset: 23
Número de campos del dataset: 5129
```

3.2) Cómo gestionarías cada uno de estos casos?

En el caso hipotético de que fuera así, el método más sencillo sería convertir dichos campos o registros en valores medios o medianos de dicha variable o atributo para no distorsionar la tendencia entre otros análisis estadísticos de los datos mencionados.

3.3) Identificación y tratamiento de valores extremos

En este dataset no se encontraron valores extremos.

Los valores extremos (o “extreme scores” o “outliers”) son aquellos registros extremos según la distribución normal de una variable o población, ya sea rango por arriba o por abajo. Por lo general, se considerará “outlier” a todo aquel valor que se encuentre alejado en una medida de tres desviaciones estándar respecto la media de la muestra poblacional. En el supuesto de que existieran, éstos deberían corregirse pues de lo contrario sesgarían significativamente los cálculos y estimaciones.

Solamente en el supuesto de que sean registros anómalos propios de la muestra poblacional. Por lo que respecta a este dataset, los registros anómalos son propios.

RESPUESTA

4) Análisis de los datos

```
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#####################################################################
Mean > Cálculo de la media
#####################################################################
print("Media aritmética del conjunto de variables:",round(tcd_df['energy'].mean(),5))
#
print("Media artimética de la variable \"energy\" de todas las canciones:",round(tcd_df['energy'].mean(),5))
#
print("Media artimética de la variable \"loudness\" de todas las canciones:",round(tcd_df['loudness'].mean(),5))
```

```
Media aritmética del conjunto de variables:
album_popularity        33.21525
danceability             0.52034
energy                   0.76898
loudness                -8.48603
speechiness              0.04939
acousticness             0.14926
instrumentalness         0.23347
liveness                 0.40388
valence                  0.50327
tempo                  129.40467
duration_ms         291956.36323
time_signature           3.95516
track_popularity        22.94170
dtype: float64

Media artimética de la variable "energy" de todas las canciones: 0.76898

Media artimética de la variable "loudness" de todas las canciones: -8.48603
```

RESPUESTA

4) Análisis de los datos

```
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#####################################################################
#Median > Cálculo de la mediana
#####################################################################
print("\nMediana del conjunto de variables:")
print(round(tcd_df.mean(),5))
#
print("Mediana de la variable \"energy\" de todas las canciones:",round(tcd_df['energy'].median(),5))
#
print("Mediana de la variable \"loudness\" de todas las canciones:",round(tcd_df['loudness'].median(),5))
```

```
Mediana del conjunto de variables:
album_popularity        33.21525
danceability             0.52034
energy                   0.76898
loudness                -8.48603
speechiness              0.04939
acousticness             0.14926
instrumentalness         0.23347
liveness                 0.40388
valence                  0.50327
tempo                  129.40467
duration_ms         291956.36323
time_signature           3.95516
track_popularity        22.94170
dtype: float64
Mediana de la variable "energy" de todas las canciones: 0.805
Mediana de la variable "loudness" de todas las canciones: -7.727
```

RESPUESTA

4) Análisis de los datos

```
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#####################################################################
#Quantiles > Cálculo de los cuartiles
#####################################################################
print("\nCuartiles del conjunto de variables:")
print(tcd_df.quantile([0.25, 0.5, 0.75, 1]),"\n")
#
print("\nCuartiles de la variable \"energy\" de las canciones:\n")
print(tcd_df['energy'].quantile([0.25, 0.5, 0.75, 1]),"\n")
#
print("\nCuartiles de la variable \"loudness\" de las canciones:\n")
print(tcd_df['loudness'].quantile([0.25, 0.5, 0.75, 1]),"\n")
```

```
Cuartiles del conjunto de variables:
      album_popularity  danceability  ...  time_signature  track_popularity
0.25              28.0         0.431  ...             4.0              17.0
0.50              34.0         0.528  ...             4.0              22.0
0.75              38.0         0.608  ...             4.0              28.0
1.00              52.0         0.848  ...             5.0              62.0

[4 rows x 13 columns]


Cuartiles de la variable "energy" de las canciones:

0.25    0.6745
0.50    0.8050
0.75    0.9000
1.00    0.9980
Name: energy, dtype: float64


Cuartiles de la variable "loudness" de las canciones:

0.25   -11.3560
0.50    -7.7270
0.75    -5.7505
1.00    -1.3780
Name: loudness, dtype: float64
```

RESPUESTA

4) Análisis de los datos

```
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#####################################################################
#Standard Deviation > Cálculo de la desviación estándar
#####################################################################
print("\nDesviación del conjunto de variables:\n"); print(round(tcd_df.std(),5))
#
print("Desviación estándar de la variable \"energy\":",round(tcd_df['energy'].std(),5))
#
print("Desviación estándar de la variable \"loudness\":",round(tcd_df['loudness'].std(),5))
```

```
Desviación del conjunto de variables:

album_popularity        10.63770
danceability             0.13588
energy                   0.15728
loudness                 3.84706
speechiness              0.03121
acousticness             0.18274
instrumentalness         0.29534
liveness                 0.32681
valence                  0.21058
tempo                   29.39382
duration_ms         118581.74275
time_signature           0.28118
track_popularity        10.09287
dtype: float64
Desviación estándar de la variable "energy": 0.15728
Desviación estándar de la variable "loudness": 3.84706
```

RESPUESTA

4) Análisis de los datos

```
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#####################################################################
#Variance > Cálculo de la varianza
#####################################################################
print("\nVarianza del conjunto de variables:\n"); print(round(tcd_df.var(),5))
#
print("\nVarianza de la variable \"energy\":",round(tcd_df['energy'].var(),5))
#
print("\nVarianza de la variable \"loudness\":",round(tcd_df['energy'].var(),5))
```

```
Varianza del conjunto de variables:

album_popularity    1.131607e+02
danceability        1.846000e-02
energy              2.474000e-02
loudness            1.479987e+01
speechiness         9.700000e-04
acousticness        3.339000e-02
instrumentalness    8.723000e-02
liveness            1.068100e-01
valence             4.434000e-02
tempo               8.639968e+02
duration_ms         1.406163e+10
time_signature      7.906000e-02
track_popularity    1.018660e+02
dtype: float64

Varianza de la variable "energy": 0.02474

Varianza de la variable "loudness": 0.02474
```

RESPUESTA

4) Análisis de los datos

```
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#####################################################################
#describe() > Cálculo de datos estadísticos
#####################################################################
print("\nDatos estadísticos del conjunto de variables:\n")
print(tcd_df.describe())
#
print("\nDatos estadísticos de la variable \"energy\":\n")
print(tcd_df['energy'].describe())
#
print("\nDatos estadísticos de la variable \"loudness\":\n")
print(tcd_df['loudness'].describe())
```

```
Datos estadísticos del conjunto de variables:

       album_popularity  danceability  ...  time_signature  track_popularity
count        223.000000    223.000000  ...      223.000000        223.000000
mean          33.215247      0.520341  ...        3.955157         22.941704
std           10.637700      0.135878  ...        0.281178         10.092867
min           11.000000      0.175000  ...        1.000000          0.000000
25%           28.000000      0.431000  ...        4.000000         17.000000
50%           34.000000      0.528000  ...        4.000000         22.000000
75%           38.000000      0.608000  ...        4.000000         28.000000
max           52.000000      0.848000  ...        5.000000         62.000000

[8 rows x 13 columns]

Datos estadísticos de la variable "energy":

count    223.000000
mean       0.768982
std        0.157281
min        0.284000
25%        0.674500
50%        0.805000
75%        0.900000
max        0.998000
Name: energy, dtype: float64

Datos estadísticos de la variable "loudness":

count    223.000000
mean      -8.486027
std        3.847060
min      -24.265000
25%      -11.356000
50%       -7.727000
75%       -5.750500
max       -1.378000
Name: loudness, dtype: float64
[Finished in 0.797s]
```

RESPUESTA

4) Análisis de los datos

4.1) Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

Seleccionaremos las variables “energy” y “loudness”.

4.2) Comprobación de la normalidad y homogeneidad de la varianza.

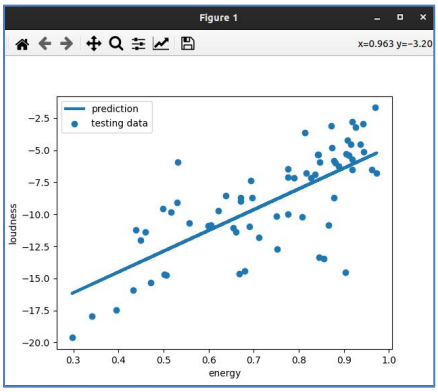
Se puede comprobar en los gráficos como las 2 variables estudiadas tienen una correlación positiva. Así pues, respecto a la normalidad podemos observar en los histogramas como la propia estructura de los datos tiene una fuerte tendencia y la distribución de la misma es creciente en relación a la frecuencia sin influir valores extremos. Por otra parte, por lo que respecta a la homogeneidad de la varianza podemos observar en los box-plots como la variación de los datos es heterogenera.

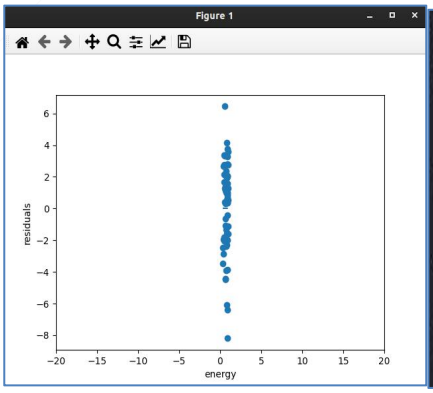
RESPUESTA

4) Análisis de los datos

4.3) Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

```
import pandas as pd; import numpy as np; import matplotlib as mpl; import matplotlib.pyplot as plt
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#########################################################################
#Evaluating model > Realizamos un análisis de las variaciones
#########################################################################
from sklearn.metrics import mean_squared_error
X = tcd_df[['energy']]; Y = tcd_df['loudness']
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, random_state=1)
from sklearn.linear_model import LinearRegression
model = LinearRegression(); model.fit(X_train, Y_train)
new_RM = np.array([6.5]).reshape(-1,1) # make sure it's 2d
y_test_predicted = model.predict(X_test)
plt.scatter(X_test, Y_test,label='testing data'); plt.plot(X_test, y_test_predicted,label='prediction', linewidth=3)
plt.xlabel('energy'); plt.ylabel('loudness'); plt.legend(loc='upper left'); plt.show()
#########################################################################
#Residuals > Analizamos los residuos
#########################################################################
residuals = Y_test - y_test_predicted
# plot the residuals
plt.scatter(X_test, residuals)
# plot a horizontal line at y = 0
plt.hlines(y = 0, xmin = X_test.min(), xmax=X_test.max(),linestyle='--')
# set xlim
plt.xlim((-20, 20)); plt.xlabel('energy'); plt.ylabel('residuals'); plt.show()
#########################################################################
#Mean Squared Error > Analizamos cuánto explicativas son las variables
#########################################################################
#################################
print("MSE (option1): ",(residuals**2).mean())
print("MSE (option2): ",mean_squared_error(Y_test, y_test_predicted),"\n")
#########################################################################
#R-squared
#########################################################################p
rint("RS (explained):",round(model.score(X_test, Y_test)*100,2),"%\n") # Model explanation variability
print("Model variation:",round(((Y_test-Y_test.mean())**2).sum(),2))
print("Total variation:",round((residuals**2).sum(),2))
```






```
MSE (option1):  7.311897909369409
MSE (option2):  7.311897909369409

RS (explained): 55.42 %

Model variation: 1098.85
Total variation: 489.9
[Finished in 84.508s]
```

RESPUESTA

5) Representación de los resultados

5.1) Tablas

```
import pandas as pd
import numpy as np
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#############################################################################
#Aggregation
#############################################################################
print("\nÁlbumes según la medias de las variables \"energy\" y \"loudness\":\n")
print(tcd_df.groupby('album_name').agg({'energy':[np.mean],'loudness':[np.mean]}))
```

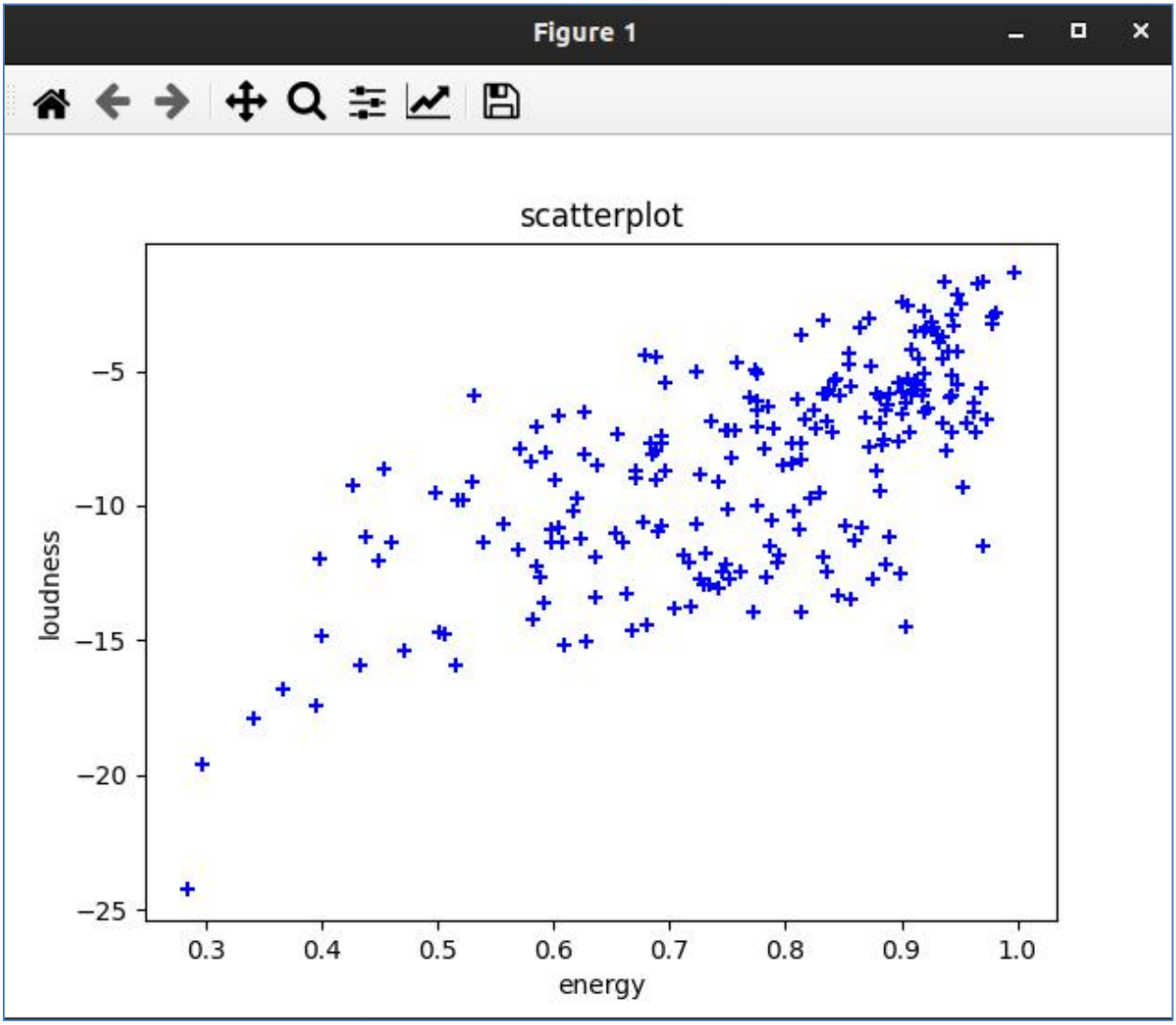
```
Álbumes según la medias de las variables "energy" y "loudness":

                                                energy   loudness
                                                  mean       mean
album_name
4:13 Dream                                    0.883846  -3.061308
Bestival Live 2011                            0.874800  -5.977250
Bloodflowers                                  0.768000  -6.363111
Concert - The Cure Live                       0.790900 -13.349100
Disintegration (Deluxe Edition [Remastered])  0.711400  -8.595750
Hypnagogic States                             0.851667  -5.411167
Kiss Me Kiss Me Kiss Me                       0.785900  -7.554100
Mixed Up (Remastered 2018 / Deluxe Edition)   0.776300  -8.584050
Paris                                         0.599833 -14.286167
Pornography                                   0.824500  -6.935875
Show                                          0.730722 -13.321722
The Cure                                      0.920000  -4.048182
The Head On The Door                          0.683000 -11.811600
The Top                                       0.715750  -8.185500
Wild Mood Swings                              0.804214  -4.509643
Wish                                          0.635250 -13.244583
[Finished in 2.587s]
```

RESPUESTA

5) Representación de los resultados

5.2) Gráficas

```
import pandas as pd
import numpy as np; import matplotlib as mpl; import matplotlib.pyplot as plt
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#########################################################################
#scatterplot > Observamos la correlación entre las variables "\energy"\ y "\loudness"\
#########################################################################
plt.scatter(tcd_df['energy'], tcd_df['loudness'],marker='+',color='b')
plt.xlabel('energy'); plt.ylabel('loudness'); plt.title('scatterplot')
plt.show()
```

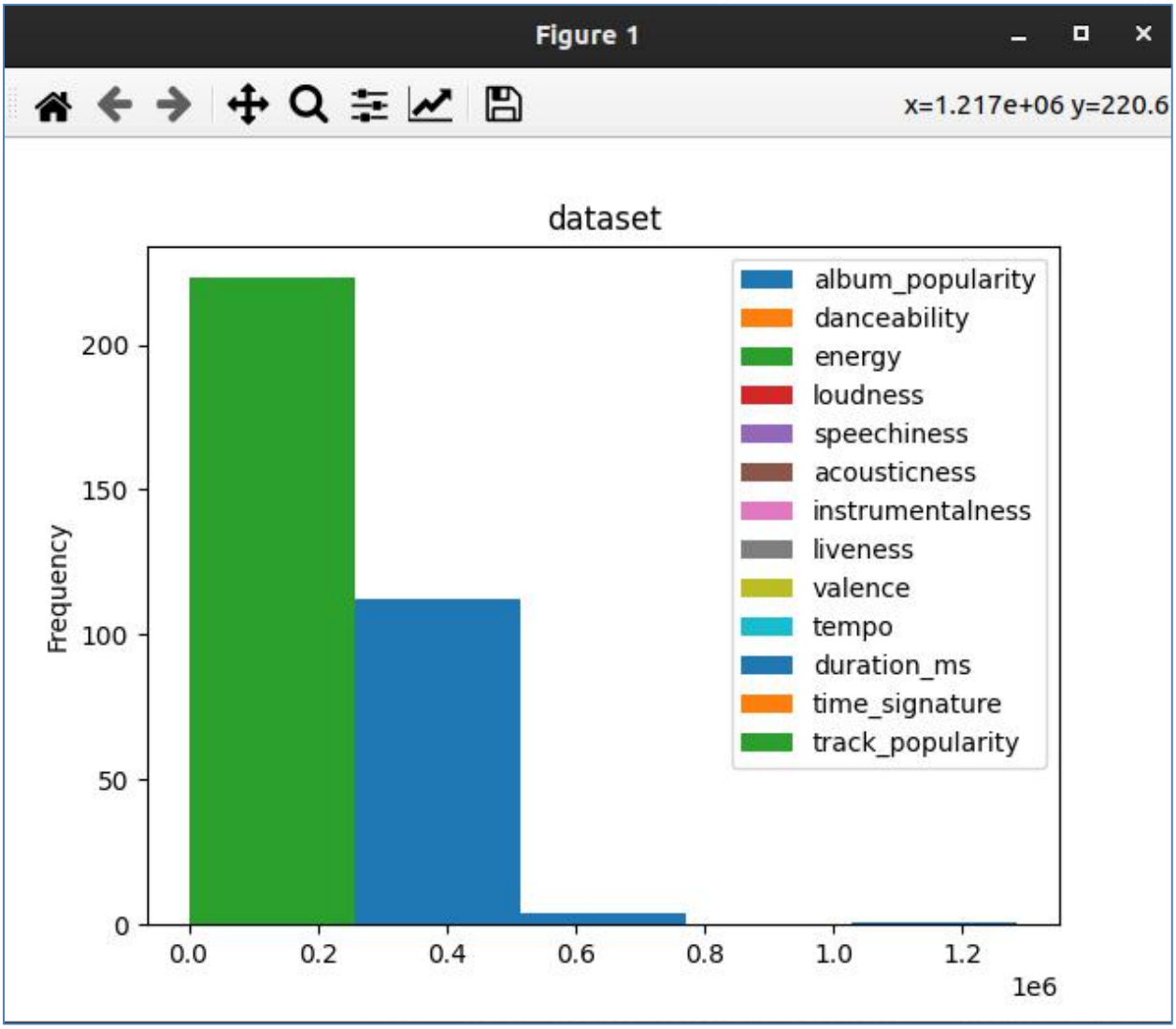


RESPUESTA

5) Representación de los resultados

5.2) Gráficas

```
import pandas as pd
import numpy as np; import matplotlib as mpl; import matplotlib.pyplot as plt
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#####################################################################
#Histogram > Observamos la evolución de las variables
#####################################################################
tcd_df.plot(kind='hist',title = 'dataset',bins=5)
plt.show()
```

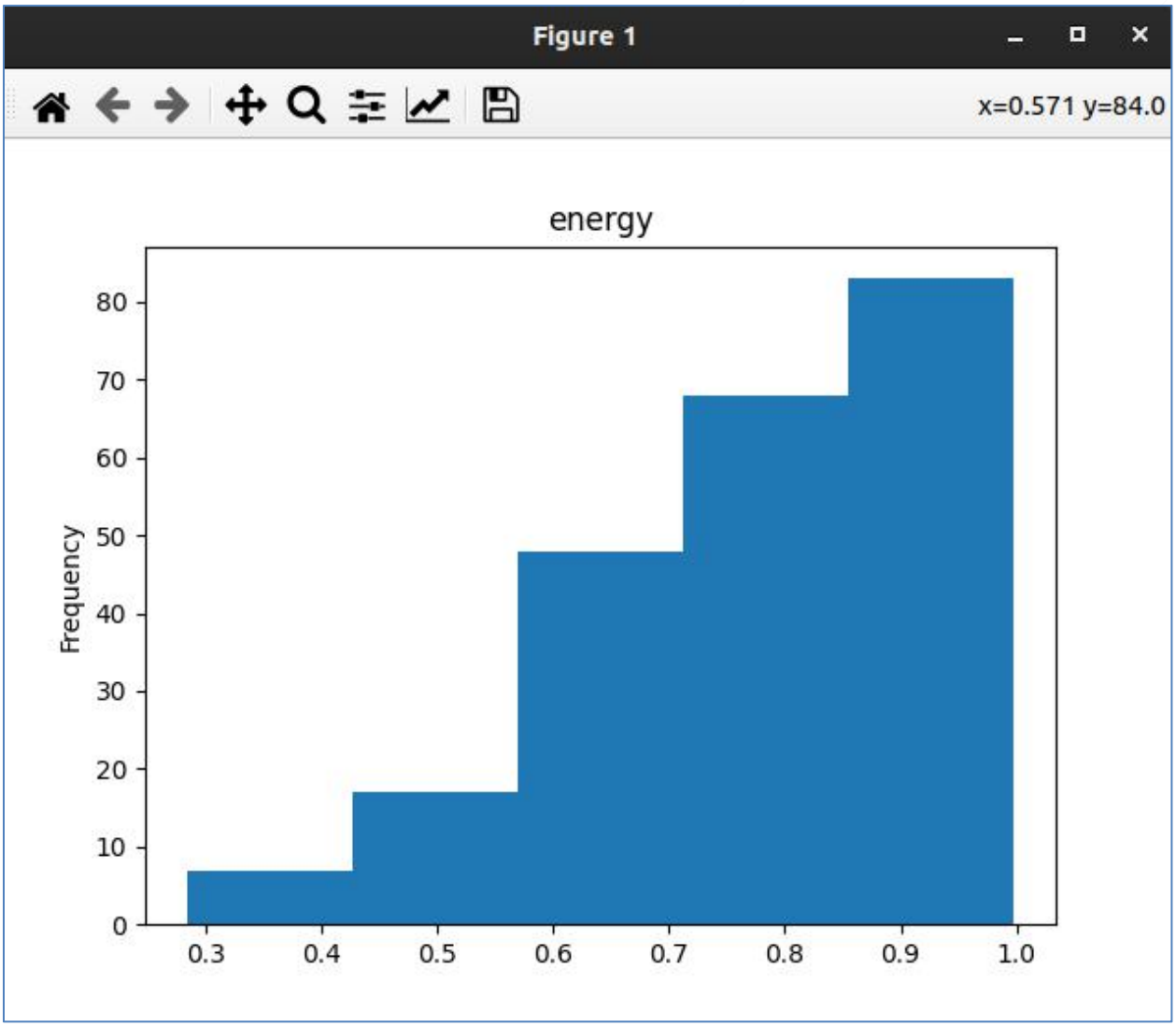


RESPUESTA

5) Representación de los resultados

5.2) Gráficas

```
import pandas as pd
import numpy as np; import matplotlib as mpl; import matplotlib.pyplot as plt
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#################################################################################
#Histogram > Observamos la evolución de las variables
#################################################################################
tcd_df['energy'].plot(kind='hist',title = 'energy',bins=5)
plt.show()
```

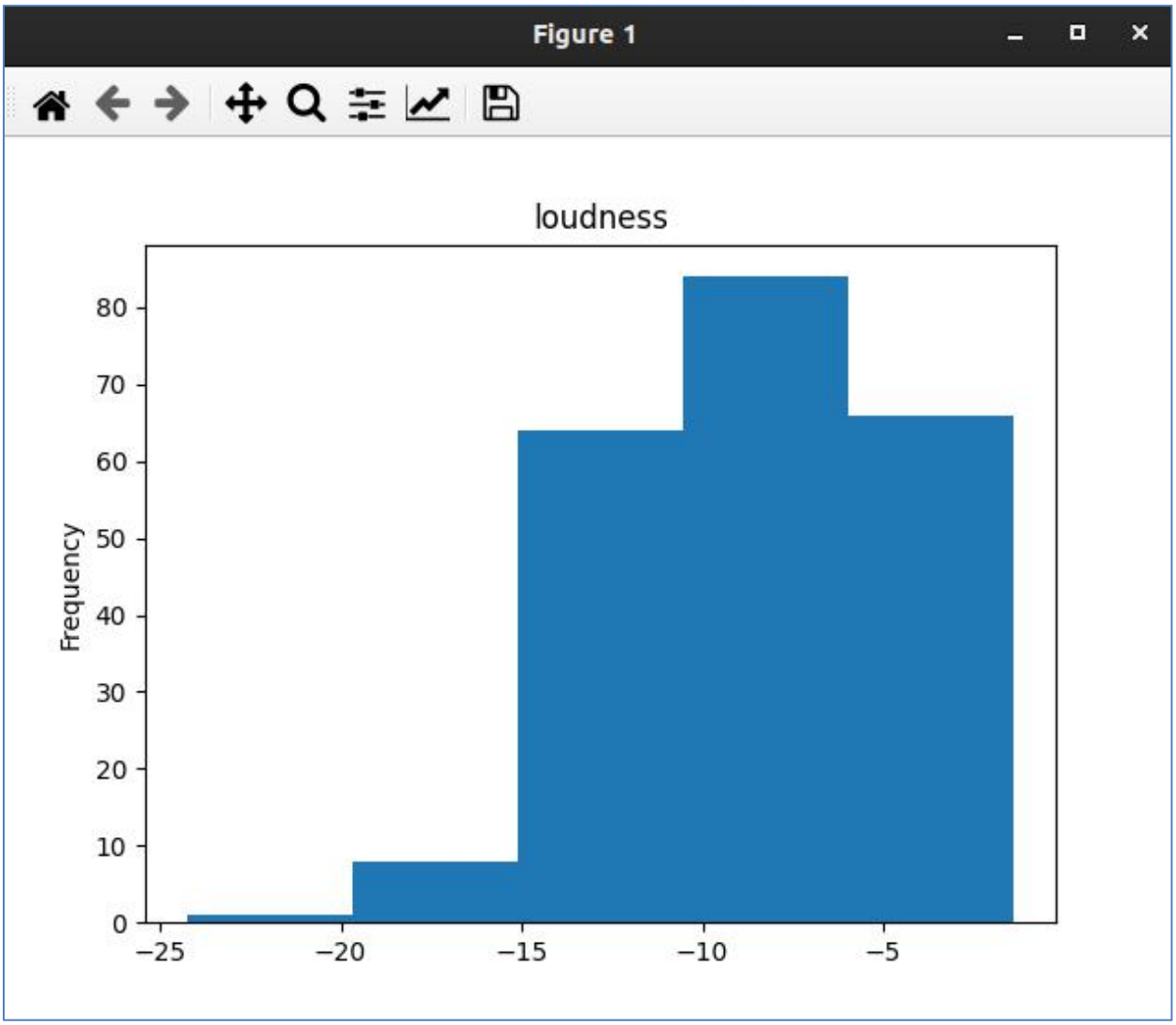


RESPUESTA

5) Representación de los resultados

5.2) Gráficas

```
import pandas as pd
import numpy as np; import matplotlib as mpl; import matplotlib.pyplot as plt
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#############################################################################
#Histogram > Observamos la evolución de las variables
#############################################################################
tcd_df['loudness'].plot(kind='hist',title = 'loudness',bins=5)
plt.show()
```

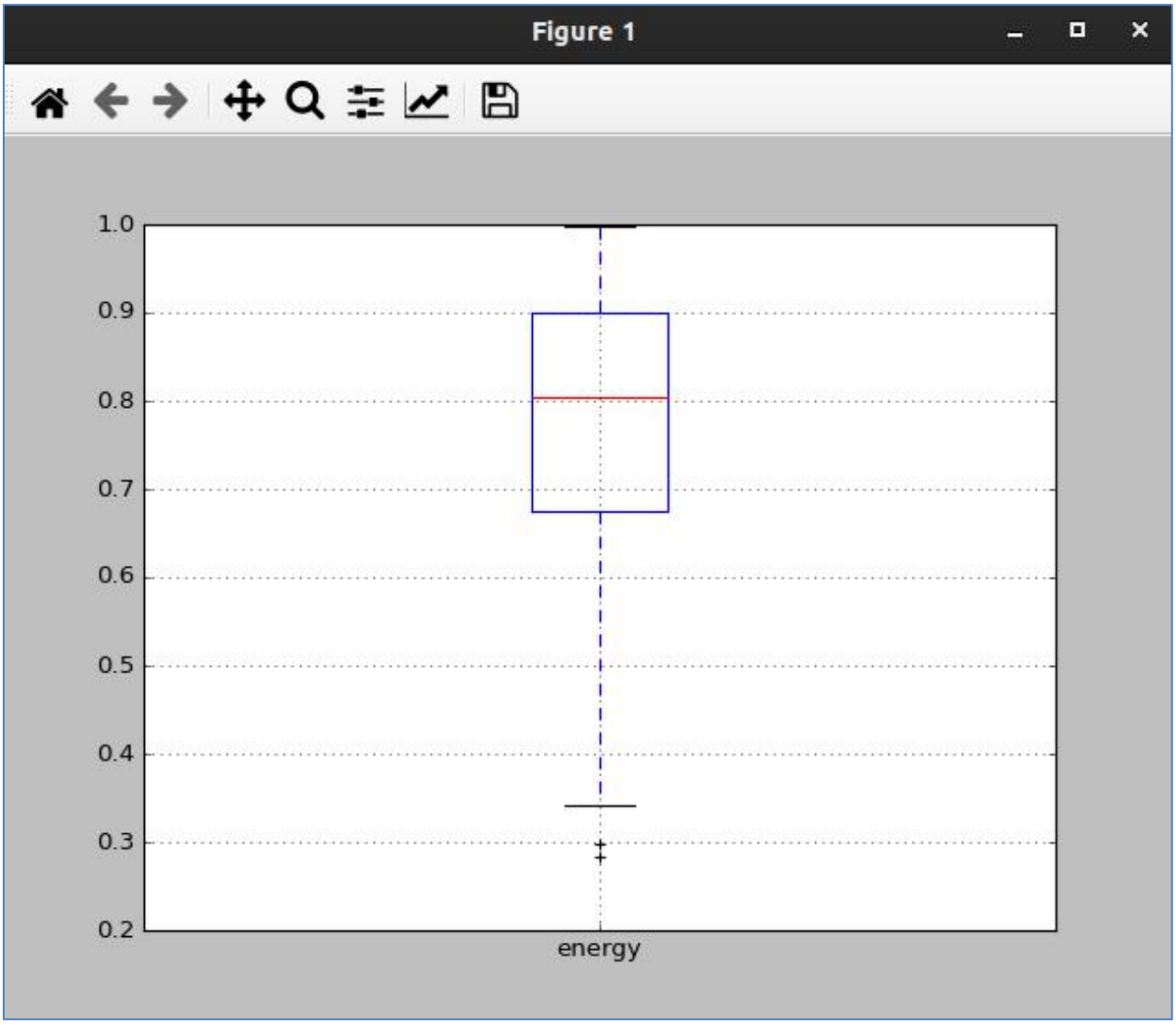


RESPUESTA

5) Representación de los resultados

5.2) Gráficas

```
import pandas as pd
import numpy as np; import matplotlib as mpl; import matplotlib.pyplot as plt
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#########################################################################
#Boxplot > Observamos la variación de las variables
#########################################################################
tcd_df['energy'].describe(); plt.style.use('classic'); tcd_df.boxplot(column='energy');
plt.show()
```

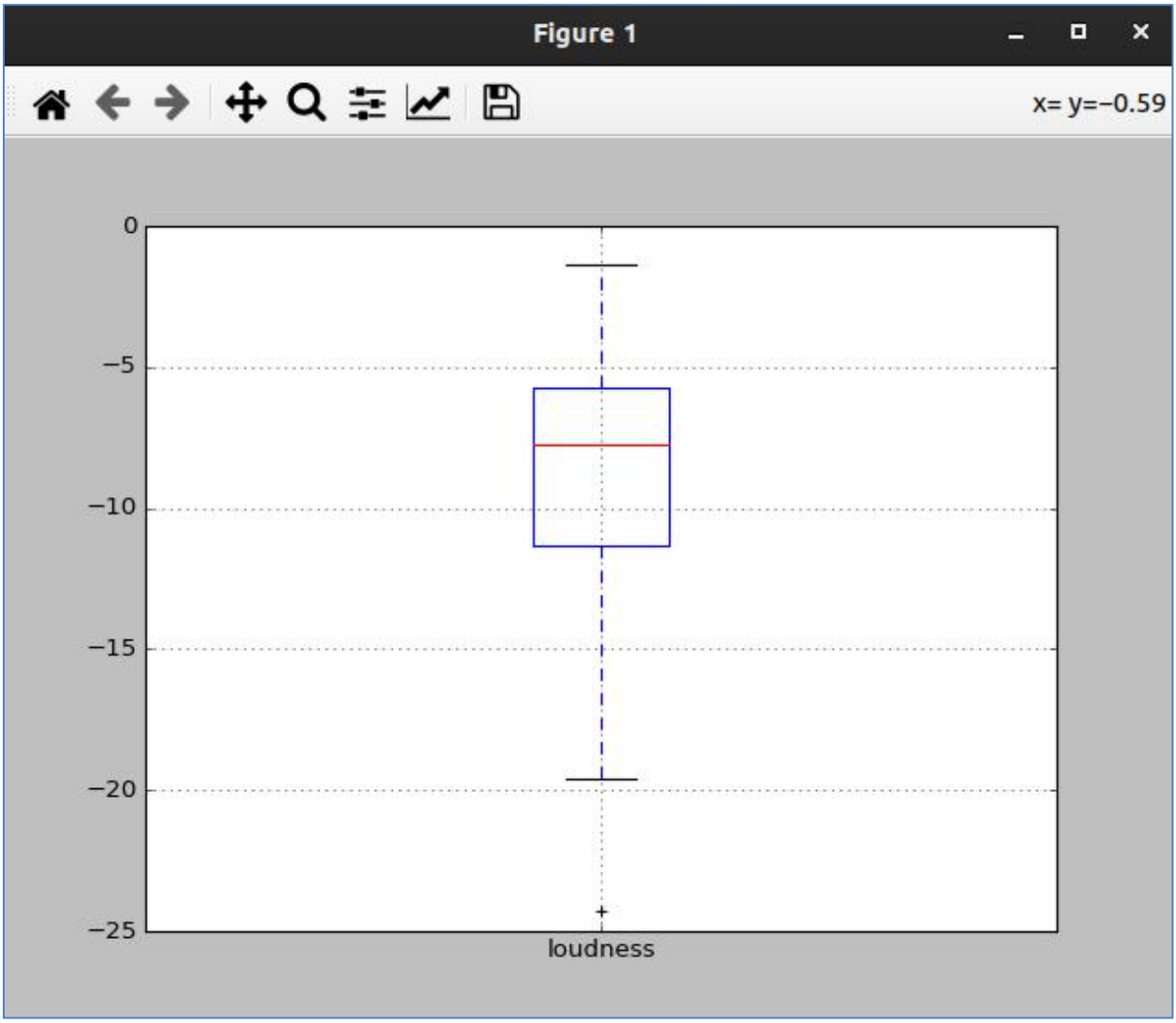


RESPUESTA

5) Representación de los resultados

5.2) Gráficas

```
import pandas as pd
import numpy as np; import matplotlib as mpl; import matplotlib.pyplot as plt
tcd_df = pd.read_csv('thecure_discography.csv', index_col=0)
#
#########################################################################
#Boxplot > Observamos la variación de las variables
#########################################################################
tcd_df['energy'].describe(); plt.style.use('classic'); tcd_df.boxplot(column='energy');
plt.show()
```



RESPUESTA

6) Resolución del problema

6.1) Cuáles son las conclusiones?

Relacionando las variables “energy” y “loudness” podemos concluir que en conjunto aquellas canciones más enérgicas o animadas son precisamente (y lógicamente) aquellas canciones más ruidosas o sonoras.

6.2) Los resultados permiten responder al problema?

En esencia, así es.

7) Código

7.1) Adjuntar el código Python

Adjuntado archivo con extensión “.py” con el conjunto de de código Python utilizado.