

Lista de Ejercicios de Programación: #06

01 - Wine Dataset (PCD33)



Escriba una función que permita leer el dataset `wine` considerando las siguientes columnas:

<https://archive.ics.uci.edu/ml/datasets/Wine>

- `Wine_class`
- `Alcohol`
- `Malic_acid`
- `Ash`
- `Alcalinity_ash`
- `Magnesium`
- `Total_phenols`
- `Flavanoids`
- `Nonflavanoid_phenols`
- `Proanthocyanins`
- `Color_intensity`
- `Hue`
- `OD280_OD315_diluted_wines`
- `Proline`

Deberá devolver solo los casos en los que el alcohol (`Alcohol`) sea mayor que 13.

La cabecera deberá ser la siguiente:

```
leer_wine_alcohol_mayor_13(filename)
```

Lo que deberá retornar un dataframe de pandas con el dataset.

Para realizar pruebas puede usar

```
filename = 'https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data'
```

El nombre del archivo: funciones.py

Link: <https://grader.labs.org.pe/web/project/997>

02 - Wine Quality - Red (PCD34)



Escriba una función que permita leer el dataset **wine quality (red)** <https://archive.ics.uci.edu/ml/datasets/Wine+Quality> considerando las siguientes columnas:

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol
- quality

Note que el separador es punto y coma, por lo que tendrá que usar el parámetro `sep=";"` — deberá devolver solo los casos en los que la calidad (`quality`) sea mayor que 3.

La cabecera deberá ser la siguiente:

```
leer_red_wine_quality_mayor_3(filename)
```

Lo que deberá retornar un dataframe de pandas con el dataset.

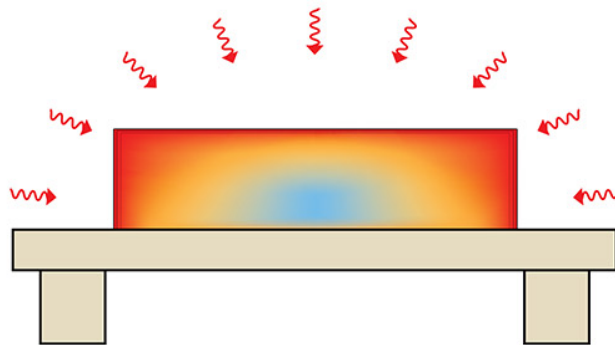
Para realizar pruebas puede usar

```
filename = 'https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv'
```

El nombre del archivo: `funciones.py`

Link: <https://grader.labs.org.pe/web/project/998>

03 - Lectura del dataset Annealing (PCD35)



Escriba una función en el lenguaje Python 3.x que lea el archivo csv del dataset **Annealing**¹. El nombre del archivo a leer será el único parámetro de esta función (para realizar pruebas puede usar: <https://archive.ics.uci.edu/ml/machine-learning-databases/annealing/anneal.data>). Los nombres de las columnas deberán ser los siguientes (puede basarse en la información de `anneal.names`²). La lectura debe ser tal que los valores denotados con `?` se interpreten como datos faltantes.

¹ <https://archive.ics.uci.edu/ml/datasets/Annealing>

² <https://archive.ics.uci.edu/ml/machine-learning-databases/annealing/anneal.names>

family	non-ageing	bw_me	corr	thick
product-type	surface-finish	bl	blue_bright_varn_clean	width
steel	surface-quality	m	lustre	len
carbon	enamelability	chrom	jurofm	oil
hardness	bc	phos	s	bore
temper_rolling	bf	cbond	p	packing
condition	bt	marvi	shape	class_value
formability		exptl		
strength		ferro		

La función deberá tener la siguiente cabecera:

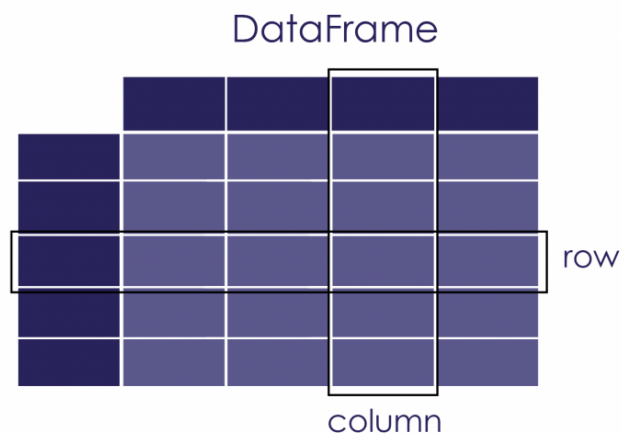
```
lectura_annealing(filename)
```

La función deberá retornar el `DataFrame` procesado.

El nombre del programa debe ser: `funciones.py`

Link: <https://grader.labs.org.pe/web/project/999>

04 - Generando un DataFrame (PCD36)



Escriba una función en el lenguaje Python 3.x que retorne un `DataFrame` de `pandas` con `M` filas y `N` columnas a partir de una matriz de unos. Los nombres de las columnas a letras: `'a'`, `'b'`, `'c'`... Si el número `N` fuera mayor a la cantidad de letras en el alfabeto del inglés, los nombres de las columnas seguirán el siguiente patrón

```
... 'z', 'aa', 'ab', 'ac', ..., 'az', 'ba', 'bb',  
    'bc', ..., 'bz', 'ca', ...
```

Asuma que $N < 300$. La función deberá tener la siguiente cabecera:

```
generar_dataframe(M, N)
```

La función deberá retornar el `DataFrame` generado.

El nombre del programa debe ser: `funciones.py`

Link: <https://grader.labs.org.pe/web/project/1000>