

Buenos días a quien corresponda:

La actividad consiste en:

Descargue del internet el archivo "Iris.csv" e impórtelo a Python de manera que pueda contestar (mediante el código respectivo) a las preguntas siguientes:

El archivo se descargó desde kaggle y se importó con la función `pd.read_csv()`, para esto antes se debe de importar la librería pandas y ponerle la abreviatura `pd`

```
import pandas as pd
```

```
iris = pd.read_csv("Iris.csv")  
iris.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

1) Determine en una sola tabla la media, desviación estándar y la mediana (percentil 50) para el ancho del pétalo por cada tipo de flor para aquellos registros cuyo ancho del sépalo es mayor o igual a 3.5 y el largo del sépalo es mayor que 5.

2) Calcule el número de registros que se utilizaron para obtener los resultados del problema 1.

Estos dos pasos se pueden hacer en una sola línea, poniendo los filtros, agrupando y agregando media, mediana, desviación y el conteo, este último nos indicará cuántos datos de cada especie hay 20 y 3 respectivamente, dando un total de 23 registros

```
iris[(iris["SepalWidthCm"] >= 3.5) & (iris["S
```

	PetalWidthCm			
	mean	median	std	count
Species				
Iris-setosa	0.285000	0.25	0.118210	20
Iris-virginica	2.233333	2.20	0.251661	3

3) Agregue una nueva columna al DataFrame generado al problema 1, que sea resultado de estandarizar los valores del ancho del pétalo para cada registro. Denomine a dicha columna “petal.width.st” y calcúlela de la siguiente manera:

$(\text{petal.width} - \text{Media}(\text{petal.width})) / \text{Desv. Estándar}(\text{petal.width})$

Esto es, a cada observación de la columna “petal.width” se le resta su media de columna para luego dividir entre la desviación estándar de su columna.

Para este punto, se hizo dos filtros, pues la media y la desviación es diferente dependiendo de la especie y se aplicó una función lambda, para al final unir los dos dataframes en uno solo con los datos solicitados

```
Iris2 = Iris_setosa.merge(Iris_virginica,how='outer')
Iris2
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species	petal.width.st
0	1	5.1	3.5	1.4	0.2	Iris-setosa	-0.719057
1	5	5.0	3.6	1.4	0.2	Iris-setosa	-0.719057
2	6	5.4	3.9	1.7	0.4	Iris-setosa	0.972842
3	11	5.4	3.7	1.5	0.2	Iris-setosa	-0.719057
4	15	5.8	4.0	1.2	0.2	Iris-setosa	-0.719057
5	16	5.7	4.4	1.5	0.4	Iris-setosa	0.972842
6	17	5.4	3.9	1.3	0.4	Iris-setosa	0.972842
7	18	5.1	3.5	1.4	0.3	Iris-setosa	0.126892
8	19	5.7	3.8	1.7	0.3	Iris-setosa	0.126892
9	20	5.1	3.8	1.5	0.3	Iris-setosa	0.126892
10	22	5.1	3.7	1.5	0.4	Iris-setosa	0.972842
11	28	5.2	3.5	1.5	0.2	Iris-setosa	-0.719057
12	33	5.2	4.1	1.5	0.1	Iris-setosa	-1.565007
13	34	5.5	4.2	1.4	0.2	Iris-setosa	-0.719057
14	37	5.5	3.5	1.3	0.2	Iris-setosa	-0.719057
15	41	5.0	3.5	1.3	0.3	Iris-setosa	0.126892
16	44	5.0	3.5	1.6	0.6	Iris-setosa	2.664742
17	45	5.1	3.8	1.9	0.4	Iris-setosa	0.972842
18	47	5.1	3.8	1.6	0.2	Iris-setosa	-0.719057
19	49	5.3	3.7	1.5	0.2	Iris-setosa	-0.719057
20	110	7.2	3.6	6.1	2.5	Iris-virginica	1.059626
21	118	7.7	3.8	6.7	2.2	Iris-virginica	-0.132453
22	132	7.9	3.8	6.4	2.0	Iris-virginica	-0.927173

4) Mantenga en el nuevo DataFrame del problema 3 las columnas “variety” y “petal.width.st”. Renómbrelas como “Variedad” y “Ancho Est. Pétalo”

Al no tener la columna Variety se sustituyo por species y se aplicó la función rename

```
Iris2.rename(columns = {"Species": "Variedad", "petal.width.st": "Ancho Est. Pétalo"}, inplace=True)
Iris2
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Variedad	Ancho Est. Pétalo
0	1	5.1	3.5	1.4	0.2	Iris-setosa	-0.719057
1	5	5.0	3.6	1.4	0.2	Iris-setosa	-0.719057
2	6	5.4	3.9	1.7	0.4	Iris-setosa	0.972842
3	11	5.4	3.7	1.5	0.2	Iris-setosa	-0.719057
4	15	5.8	4.0	1.2	0.2	Iris-setosa	-0.719057
5	16	5.7	4.4	1.5	0.4	Iris-setosa	0.972842
6	17	5.4	3.9	1.3	0.4	Iris-setosa	0.972842
7	18	5.1	3.5	1.4	0.3	Iris-setosa	0.126892
8	19	5.7	3.8	1.7	0.3	Iris-setosa	0.126892
9	20	5.1	3.8	1.5	0.3	Iris-setosa	0.126892
10	22	5.1	3.7	1.5	0.4	Iris-setosa	0.972842
11	28	5.2	3.5	1.5	0.2	Iris-setosa	-0.719057
12	33	5.2	4.1	1.5	0.1	Iris-setosa	-1.565007
13	34	5.5	4.2	1.4	0.2	Iris-setosa	-0.719057
14	37	5.5	3.5	1.3	0.2	Iris-setosa	-0.719057
15	41	5.0	3.5	1.3	0.3	Iris-setosa	0.126892
16	44	5.0	3.5	1.6	0.6	Iris-setosa	2.664742
17	45	5.1	3.8	1.9	0.4	Iris-setosa	0.972842
18	47	5.1	3.8	1.6	0.2	Iris-setosa	-0.719057
19	49	5.3	3.7	1.5	0.2	Iris-setosa	-0.719057
20	110	7.2	3.6	6.1	2.5	Iris-virginica	1.059626
21	118	7.7	3.8	6.7	2.2	Iris-virginica	-0.132453
22	132	7.9	3.8	6.4	2.0	Iris-virginica	-0.927173

5) Exporte el resultado del DataFrame obtenido en el problema 4 a un archivo denominado “Iris2.csv”.

Para exportarlo basta con poner to_csv al dataframe. En lugar de csv también se puede exportar a un Excel cambiando csv por Excel y la extensión del archivo por.xlsx. Si se decide esta opción se debe de considerar eliminar el índice para que no aparezca en la columna A del Excel.

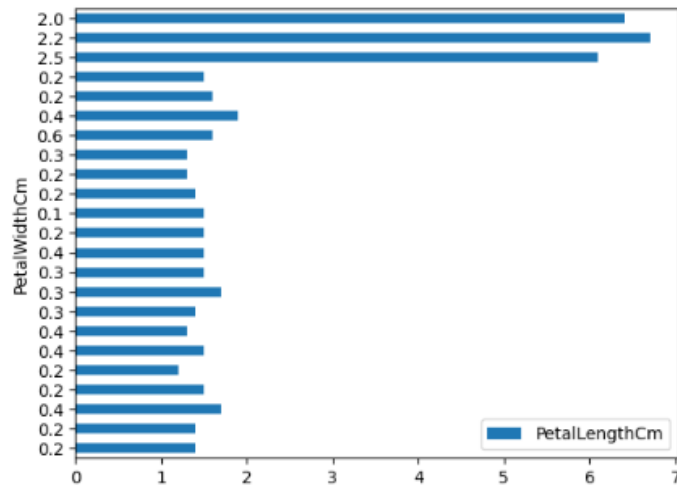
```
Iris2.to_csv = "Iris2.csv"
```

6) Investigue cómo generar un gráfico de dispersión para la base original Iris que relacione la variable “petal.width” con “petal.length”. Sugerencia: Revisar la instrucción. plot(x,y,kind)

Finalmente, con la función plot se puede graficar el dataframe sin importar otras librerías como plotly, matplotlib o seaborn, el atributo kind sirve para indicar el tipo de gráfico, de barras barh, de puntos de sipersion scatter o line entre otros. En este caso, no recomiendo line porque no es el adecuado para los datos.

```
Iris2.plot("PetalWidthCm", "PetalLengthCm", kind = "barh")
```

<Axes: ylabel='PetalWidthCm'>



```
Iris2.plot("PetalWidthCm", "PetalLengthCm", kind = "scatter")
```

<Axes: xlabel='PetalWidthCm', ylabel='PetalLengthCm'>

