



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE COMPUTACIÓN Y SIMULACIÓN DE SISTEMAS
TÓPICOS ESPECIALES I
TALLER 2



Facilitador: Tomás J. Concepción Miranda

Integrantes:

Gustavo Colucci 8-951-2191

Luis Mejía 8-949-350

Greg Torres, 8-956-675

Objetivos

Aprender las funciones y métodos fundamentales de la librería Pandas.

Indicaciones

Se debe realizar un informe en el que se detalle, para cada problema, el desarrollo de la solución. Este trabajo se puede realizar en grupos de 2 o 3 personas. **Envíe su informe en Moodle en formato PDF**, así como el **código fuente**, en el bloque correspondiente **antes de las 11:55 p.m. del día siguiente del taller**.

Rúbrica

Este laboratorio tendrá una puntuación total de 100, donde la evaluación se basada en los aspectos de excelente, bueno, regular, deficiente. Los puntos que se evaluará en la rúbrica se muestran en la tabla:

N.	Aspectos a evaluar				
1	CONTENIDO DE ACUERDO A LO SOLICITADO EN EL ENUNCIADO	Excelente (80)	Bueno (70)	Regular (50)	Deficiente (0)
	Resultados debidamente presentados (capturas de pantalla, explicación) - (80 puntos)				
2	PANTALLA DE PRESENTACIÓN - (10 puntos)	Excelente (10)	Bueno (8)	Regular (5)	Deficiente (0)
	Cumple con todos los parámetros dados en clase				
3	ENTREGA DE TRABAJO EN LA PLATAFORMA MOODLE - (10 puntos)	Excelente (10)	Bueno (8)	Regular (5)	Deficiente (0)
	Entrega a tiempo en la plataforma				

Enunciados

Información General

Pandas es una librería que provee una estructura de datos diseñada para que trabajar con datos “relacionales” o “etiquetados” sea fácil e intuitivo. Esta librería es adecuada para los siguientes tipos de datos:

- Datos tabulados con columnas heterogéneamente tipadas, como en el caso de tablas SQL u hojas de cálculo tipo Excel
- Datos de series temporales ordenadas o desordenadas
- Datos arbitrarios matriciales con etiquetas de filas y columnas
- Cualquier otro juego de datos observacionales / estadísticos.

Los datos no necesitan estar etiquetados para ser puestos en una estructura pandas. Para instalar Pandas, basta con lanzar el comando `pip install pandas`. La documentación oficial se encuentra disponible en pandas.pydata.org/docs

Problema 1: Describa el contenido de `datos1`:

Explicación: bueno para describir la tabla primero creamos una tabla y luego imprimimos con `print` y el nombre de la tabla

```
4  datos1 = {
5      "nombre": ["Pedro", "Maria", "Antonio"],
6      "cliente_desde": ["2022-01-23", "2021-11-24", "2023-06-23"],
7      "num_compras": [19,12,3],
8  }
9  print(datos1)
```

	nombre	cliente_desde	num_compras
0	Pedro	2022-01-23	19
1	Maria	2021-11-24	12
2	Antonio	2023-06-23	3

Problema 2: Cree un `DataFrame` llamado `tabla1` a partir de `datos1`. Describa la `tabla1`:

Explicación: Para crear el dataframe primero debemos crear la tabla con sus datos y esa tabla la metemos dentro de unos paréntesis del dataframe con el puntero de pandas y lo guardamos.

```

1 import pandas as pd
2 ##Problema 1 crear un data frame
3 datos1 = {
4     "nombre": ["Pedro", "Maria", "Antonio"],
5     "cliente_desde": ["2022-01-23", "2021-11-24", "2023-06-23"],
6     "num_compras": [19,12,3],
7 }
8 df_datos1 = pd.DataFrame(datos1)

```

	nombre	cliente_desde	num_compras
0	Pedro	2022-01-23	19
1	Maria	2021-11-24	12
2	Antonio	2023-06-23	3

Problema 3: Seleccione la columna num_compras e imprímala:

Explicación: este problema se puede realizar de 2 maneras

- Mediante la función series la cual va a agarrar exclusivamente las columnas num_compras y nos va a decir de qué tipo es el objeto.
- Esta segunda opción es la más básica de Python que consiste en solamente seleccionar la columna del arreglo.

```

11 ##Problema 2
12 print(pd.Series(datos1 , index = ["num_compras"]))
13 #o
14 print(df_datos1["num_compras"])
15

```

	num_compras
0	19
1	12
2	3

Problema 4: Cree un arreglo Series llamado compra_promedio con los siguientes datos:

Explicación: en este proceso usaremos nuevamente la función series que sirve para añadir datos a una tabla nueva y esta la guardamos en la tabla compra_promedio.

Á

```

16 #Problema 4
17 compra_promedio = pd.Series([9.41, 10.8, 12.74])
18 print(compra_promedio)

```

	compra_promedio
0	9.41
1	10.80
2	12.74

Problema 5: Agregue el arreglo compra_promedio a tabla1:

Explicación: En este paso, se decidió agregar una nueva columna al DataFrame tabla1. Esta columna se llama compra_promedio y la creé a partir de un arreglo Series que ya había definido anteriormente.

Al agregar esta columna, `tabla1` ahora tiene un registro adicional de información para cada cliente, que indica el promedio de sus compras.

```
34 #Problema 5 Agregue la columna compra_promedio a la tabla1
35 print("Problema 5")
36 tabla1["compra_promedio"] = compra_promedio
37 print("La tabla1 con la columna compra_promedio es: \n",tabla1)
38 print("\n")
39
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Problema 5
La tabla1 con la columna compra_promedio es:

	Nombre	cliente_desde	num_compras	compra_promedio
0	Pedro	2022-01-23	19	9.41
1	Maria	2021-11-24	12	10.80
2	Juan	20223-06-23	3	12.74

Problema 6: Imprima las estadísticas de la columna `compra_promedio`:

Explicación: Después de agregar la columna `compra_promedio`, utilicé el método `describe()`, que me proporciona información como el promedio, el valor mínimo, el valor máximo, la desviación estándar, entre otros. Esto me ayuda a tener una idea general de cómo se distribuyen los valores de compra promedio entre los clientes.

```
39
40 #Problema 6 Imprima las estadísticas de la columna compra_promedio
41 print("Problema 6")
42 print("Las estadísticas de la columna compra_promedio son: \n",tabla1["compra_promedio"].describe())
43 print("\n")
44
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Problema 6
Las estadísticas de la columna compra_promedio son:

count	3.000000
mean	10.983333
std	1.672553
min	9.410000
25%	10.105000
50%	10.800000
75%	11.770000
max	12.740000

Name: compra_promedio, dtype: float64

Problema 7: Cree un DataFrame llamado `iris` a partir del archivo `iris.csv`:

Explicación: En el problema 7, cargué el conjunto de datos `iris` desde un archivo CSV usando la función `read_csv` de `pandas`. Asigné el contenido a un DataFrame llamado `iris`, suponiendo que el archivo estaba en la ruta `"Taller#2\iris.csv"`

```

44
45 #Problema 7 Cree un DataFrame llamado iris a partir del archivo iris.csv
46 print("Problema 7")
47 iris = pd.read_csv("Taller#2\iris.csv")
48 print("El DataFrame iris es: \n",iris)
49 print("\n")
50

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Problema 7
El DataFrame iris es:

	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
..
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

[150 rows x 6 columns]

Problema 8: Imprima las columnas del DataFrame iris.

```

1 import pandas as pd
2
3 iris =pd.read_csv("./iris.csv")
4
5 # problema 8
6
7 print(iris.columns)

```

Resolución del problema 8

```

Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
      'Species'],
      dtype='object')

```

Resultados del problema 8

Explicación: Después de cargado el DataFrame de iris, usamos la función **columns**, la cual genera todo el **Índice** del **DataFrame** y muestra el tipo de datos que es.

Problema 9: Ordene el DataFrame iris por PetalLengthCm:

```

1 import pandas as pd
2
3 iris =pd.read_csv("./iris.csv")
4
5 # problema 9
6
7 iris=iris.sort_values(by=["PetalLengthCm"])
8 print(iris)

```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
22	23	4.6	3.6	1.0	0.2	Iris-setosa
13	14	4.3	3.0	1.1	0.1	Iris-setosa
14	15	5.8	4.0	1.2	0.2	Iris-setosa
35	36	5.0	3.2	1.2	0.2	Iris-setosa
36	37	5.5	3.5	1.3	0.2	Iris-setosa
..
131	132	7.9	3.8	6.4	2.0	Iris-virginica
105	106	7.6	3.0	6.6	2.1	Iris-virginica
117	118	7.7	3.8	6.7	2.2	Iris-virginica
122	123	7.7	2.8	6.7	2.0	Iris-virginica
118	119	7.7	2.6	6.9	2.3	Iris-virginica

[150 rows x 6 columns]

Explicación: Cargado el DataFrame de iris, usamos la función `sort_value()` (la cual ordena los valores por la columna que le indiquemos. En este caso se utilizó **“PetalLengthCm”**. Si no se especifica la forma determinada de ordenamiento de `sort_value()` es de forma ascendente.

```
1 import pandas as pd
2
3 iris =pd.read_csv(".\iris.csv")
4
5 #problema 10
6 |
7 Max=iris["PetalLengthCm"].max()
8 print(iris[iris["PetalLengthCm"] == Max])
```

		Id	SepallLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
118	119		7.7	2.6	6.9	2.3	Iris-virginica

Explicación: Cargado el DataFrame de iris, en una variable llamada **Max** guardaremos el valor más alto de la tabla en la columna de **“PetalLengthCm”**, esto usando la función **max()** la cual tiene que estar precedida en este caso por el nombre de la columna dentro de un []. Usando el valor máximo usamos los corchetes para escribir que se imprima el valor en la columna **“PetalLengthCm”**, que sea igual al valor de **Max**. Así obteniendo el resultado.

Problema 11: Imprima todas las entradas del DataFrame iris cuyo valor PetalLengthCm sea igual o mayor a 5.0:

```

1  import pandas as pd
2
3  iris =pd.read_csv(".\iris.csv")
4
5  #problema 11
6  print(iris[iris["PetalLengthCm"] >= 5.0])

```

PROBLEMAS	SALIDA	CONSOLA DE DEPURACIÓN	TERMINAL			
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
77	78	6.7	3.0	5.0	1.7	Iris-versicolor
83	84	6.0	2.7	5.1	1.6	Iris-versicolor
100	101	6.3	3.3	6.0	2.5	Iris-virginica
101	102	5.8	2.7	5.1	1.9	Iris-virginica
102	103	7.1	3.0	5.9	2.1	Iris-virginica
103	104	6.3	2.9	5.6	1.8	Iris-virginica
104	105	6.5	3.0	5.8	2.2	Iris-virginica
105	106	7.6	3.0	6.6	2.1	Iris-virginica
107	108	7.3	2.9	6.3	1.8	Iris-virginica
108	109	6.7	2.5	5.8	1.8	Iris-virginica
109	110	7.2	3.6	6.1	2.5	Iris-virginica
110	111	6.5	3.2	5.1	2.0	Iris-virginica
111	112	6.4	2.7	5.3	1.9	Iris-virginica
112	113	6.8	3.0	5.5	2.1	Iris-virginica
113	114	5.7	2.5	5.0	2.0	Iris-virginica
114	115	5.8	2.8	5.1	2.4	Iris-virginica
115	116	6.4	3.2	5.3	2.3	Iris-virginica
116	117	6.5	3.0	5.5	1.8	Iris-virginica
117	118	7.7	3.8	6.7	2.2	Iris-virginica
118	119	7.7	2.6	6.9	2.3	Iris-virginica
119	120	6.0	2.2	5.0	1.5	Iris-virginica
120	121	6.9	3.2	5.7	2.3	Iris-virginica
122	123	7.7	2.8	6.7	2.0	Iris-virginica
124	125	6.7	3.3	5.7	2.1	Iris-virginica
125	126	7.2	3.2	6.0	1.8	Iris-virginica

Resolución del problema 11

Explicación: usamos nuevamente los corchetes. Primero colocamos el nombre del DataFrame, luego abrimos corchete y colocamos la condición. La cual sería `iris["PetalLengthCm"] >= 5.0` y cerramos corchete. Esta condición le está diciendo al programa que solo escoja los valores que sean iguales o mayores que el 5 en la columna “PetalLengthCm”.