PROGRAMACIÓN tics100

FACULTAD DE INGENIERÍA Y CIENCIAS.
UNIVERSIDAD ADOLFO IBAÑEZ

Segundo semestre 2020

Pandas (I)

Análisis de Datos

- El análisis de datos es el proceso de evaluar datos utilizando herramientas analíticas y estadísticas para descubrir información útil y ayudar en la toma de decisiones de negocios.

Pandas

Pandas es un módulo (de código abierto) que proporciona:

- estructuras de datos de alto rendimiento y
- herramientas de análisis de datos

para el lenguaje de programación Python.

```
#importar pandas
import pandas as pd
```

Pandas

Pandas está basado en otros módulos, tales como Numpy, y usa principalmente dos estructuras de datos:

- **Series**
- **DataFrames**

Nos enfocaremos en el último. Un DataFrame es una matriz (arreglo bidimensional) etiquetada con columnas de tipos potencialmente diferentes.

	NAME	AGE	DESIGNATION	
1	a	20	VP	
2	b	27	CEO	
3	c	35	CFO	
4	d	55	VP	
5	e	18	VP	
6	f	21	CEO	
7	g	35	MD	

¿Para qué sirve?

Un dataframe nos permite:

- Cargar datos de otra fuente de datos (como un CSV)
- Mantener datos en forma ordenada
- Acceder por nombre o por índice (posición)
- Realizar operaciones sobre datos (como Numpy)
- Entre otras posibilidades

Ejemplo

- Para aprender Pandas de forma más práctica, iremos paso a paso usando datos de los juegos olímpicos obtenidos de https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results
  - Una copia de los datos se encuentra en webcursos para que los bajen
- Lo que haremos es:
  - Cargar los datos
  - Analizar un dataframe
  - Acceder a datos
  - Realizar filtros
  - Obtener estadísticas

Carga de Datos

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

print(df.describe())

print(df.head(10))
```

read_csv() lee un archive csv y lo carga en un dataframe

describe() nos permite conocer datos generales del dataframe

head(n) nos muestra los primeros n registros

	ID	Age	…
count	271116.000000	261642.000000	…
mean	68248.954396	25.556898	…

	ID	Name	Sex	...	Sport	Event	Medal	…
0	1	A Dijiang	M	...	Basketball	Basketball Men's Basketball	NaN	…
1	2	A Lamusi	M	...	Judo	Judo Men's Extra-Lightweight	NaN	…
2	3	Gunnar Nielsen Aaby	M	...	Football	Football Men's Football	NaN	…
3	4	Edgar Lindenau Aabye	M	...	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold	…
4	5	Christine Jacoba Aaftink	F	...	Speed Skating	Speed Skating Women's 500 metres	NaN	…

[5 rows x 15 columns]

UNIVERSIDAD ADOLFO IBÁÑEZ

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

print(df.iloc[0])
```

Muestra el primer registro del listado de atletas

```
print(df.iloc[0,1])
```

*Muestra el campo 2 del primer registro (**A Dijiang**)*

```
ID                                      1
Name                            A Dijiang
Sex                                     M
Age                                    24
Height                                180
Weight                                 80
Team                               China
NOC                                  CHN
Games                        1992 Summer
Year                                 1992
Season                             Summer
City                            Barcelona
Sport                          Basketball
Event     Basketball Men's Basketball
Medal                                NaN
Name: 0, dtype: object
```

UNIVERSIDAD ADOLFO IBÁÑEZ

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

print(df['Name'])
```

Podemos obtener todos los elementos de una columna usando su nombre

```
print(df['Name'][0])
```

Muestra el primer registro del listado anterior

```
0                                  A Dijiang
1                                   A Lamusi
2                        Gunnar Nielsen Aaby
3                       Edgar Lindenau Aabye
4                    Christine Jacoba Aaftink
5                    Christine Jacoba Aaftink
6                    Christine Jacoba Aaftink
7                    Christine Jacoba Aaftink
…
```

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

data = df[['Name', 'Team']]
```

Podemos obtener más de una columna

```
print(data.iloc[0])
```

Muestra el primer registro del listado anterior

```
Name    A Dijiang
Team         China
Name: 0, dtype: object
```

Nota

¿Por qué cuando realizamos la consulta de 1 columna pudimos ir al primer registro haciendo [0] directamente, en cambio en la consulta con 2 columnas tuvimos que usar .iloc[0]?

- Cuando consultamos por 1 columna obtenemos una serie, el equivalente a un arreglo de 1 dimensión ☐ funciona como una lista
- En cambio, al usar 2 o más columnas, obtenemos un dataframe ☐ debemos usarlo como dataframe ☺

Estadísticas simples

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

print(df.shape[0])
```

shape() nos permite conocer las dimensiones… ¡igual que en Numpy!

```
unique_athletes = df['Name'].unique()
```

unique() nos permite eliminar los elementos duplicados

```
print(list(unique_athletes))
```

Para mostrar todos los datos en una colección una dimension podemos usar list()

```
print(len(unique_athletes))
```

Al igual que Numpy, podemos usar len() para contar elementos en una colección de una dimensión

Filtros

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

gold_winners = df[df['Medal'] == 'Gold']
```

Podemos colocar condiciones para buscar en vez de usar posiciones

```
unique_gold_winners = gold_winners['Name'].unique()
print(unique_medal_winners)
```

Filtros

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

medal_winners = df[(df['Medal'] == 'Gold') | (df['Medal'] ==
'Silver') | (df['Medal'] == 'Bronze')]
```

Podemos colocar multiples condiciones usando & (and), | (or) y separando con paréntesis las condiciones

```
unique_medal_winners = medal_winners['Name'].unique()
print(unique_medal_winners)
```

Filtros

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

medal_winners = df[df['Medal'].fillna('None') != 'None']
```

Podemos usar la función fillna() para darle un valor a los que no tienen medulla y luego comparar contra ese valor

```
unique_medal_winners = medal_winners['Name'].unique()
print(unique_medal_winners)
```

¿Es lo mismo?

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

opcion_1 = df[df['Medal'].fillna('None') != 'None']

opcion_2 = df[(df['Medal'] == 'Gold') | (df['Medal'] ==
'Silver') | (df['Medal'] == 'Bronze')]

print(opcion_1.equals(opcion_2))
```

La función equals() permite ver si 2 dataframes tienen los mismos datos

Ejercicio

Tiempo : 20 minutos

Los datos de los juegos olímpicos contiene las siguientes columnas:

- ID - ID único para cada atleta
- Name - El nombre del atleta
- Sex - Género (F o M)
- Age - Edad del atleta al momento de los juegos
- Height - Altura en centímetros
- Weight - Peso en kilogramos
- Team - Equipo
- NOC - National Olympic Committee – código de 3 letras
- Games - Año y temporada
- Year - Año
- Season - Temporada (Summer o Winter)
- City - Ciudad anfitriona
- Sport - Deporte
- Event - Evento
- Medal - Medalla obtenida (Gold, Silver, Bronze, o NA si no obtuvo)

Muestre los nombres (sin repetir) de los atletas para cada una de las condiciones siguientes:

1. **Atletas de género femenino que hayan participado en los juegos de verano**
2. **Atletas de más de 190 centímetros o 90 kilos de peso**

Ejercicio

Tiempo : 20 minutos

```
import pandas as pd

df = pd.read_csv('athlete_events.csv')

women_summer = df[(df['Sex']=='F') & (df['Season'] == 'Summer')]
unique_women_summer = women_summer['Name'].unique()
print(unique_women_summer)

big_athletes = df[(df['Weight']>=90.0) | (df['Height'] >= 190)]
unique_big_athletes = big_athletes['Name'].unique()
print(unique_big_athletes)
```

Ejercicio

Tiempo : 20 minutos

Más consultas

1. Muestre los nombres y deporte de los atletas de Chile que hayan participado en los juegos de 1996
2. Muestre los deportes que se han practicado en los juegos olímpicos de invierno desde 1980