Pandas

Introducción

Tres estructuras de datos diferentes:

* Series: Estructura de una dimensión.
* DataFrame: Estructura de dos dimensiones (tablas).
* Panel: Estructura de tres dimensiones (cubos).

Links importantes:

* [https://pandas.pydata.org](https://pandas.pydata.org/)
* [https://matplotlib.org](https://matplotlib.org/)
* [https://seaborn.pydata.org](https://seaborn.pydata.org/)

# 

# Serie

Matriz unidimensional etiquetada capaz de contener cualquier tipo de datos. Sus elementos tienen que ser del mismo tipo y su tamaño es inmutable aunque sí su contenido. Son estructuras similares a los arrays de una dimensión.

#Serie creada con una lista

una\_serie = pd.Series(['Banana', 'Manzana', 'Naranja', 'Durazno', 'Frutilla'], dtype='string')

#Serie creada con un diccionario

otra\_serie = pd.Series({'Banana': 4, 'Manzana': 3, 'Naranja': 2})

Métodos

* size -> número de elementos de la serie
* index -> lista con los nobres de las filas
* dtype -> tipo de datos de los elementos de la serie

Acceso a los elementos

* serie[i] -> elemento que ocupa la posición i+1 en la serie
* serie[posiciones] -> serie con los elementos que ocupan las posiciones de la lista de posiciones
* serie[nombre] -> elemento con el nombre ‘nombre’ en el índice
* serie[nombres] -> serie con los elementos correspondientes a los nombres indicados en la lista ‘nombres’ en el índice

# 

# DataFrame aka DataSet

Estructura de datos bidimensional donde cada columna es un objeto de tipo Series. Todos los datos de una misma columna son del mismo tipo y las filas son registros que pueden contener datos de distintos tipos.

Contiene dos índices: uno para las filas y otro para las columnas y se puede acceder a sus elementos mediante los nombres de las filas y las columnas.

#DataFrame a partir de una lista

df1 = pd.DataFrame([['Juan','Pérez',25,'Ingeniero'], ['Luis','Gómez',30,'Abogado'], ['Antonio','Sánchez',20,'Médico']], columns=['nombre', 'apellido','edad','profesion'])

#DataFrame a partir de un diccionario

datos = {'nombre':['Juan', 'Luis', 'Antonio'],

'apellido':['Pérez','Gómez','Sánchez'],

'edad':[25, 30, 35],

'profesion':['Ingeniero', 'Abogado', 'Médico']

}

df2 = pd.DataFrame(datos)

Métodos

* info -> n° filas, n° columnas, índices, tipo de columnas y memoria usada
* shape -> tupla con el número de filas y columnas
* size -> número de elementos
* columns -> lista con los nombres de las columnas
* index -> lista con los nombres de las filas
* dtypes -> serie con los tipos de datos de las columnas
* head(n) -> n primeras filas
* tail(n) -> n últimas filas

Acceso a los datos

Mediante posiciones:

* df.iloc[i, j] -> elemento en la fila i y la columna j
* df.iloc[filas, columnas] -> dataframe con los elementos en las filas de la lista ‘filas’ y de las columnas de la lista ‘columnas’
* df.iloc[i] -> serie con los elementos de la fila i

Mediante nombres:

* df.loc[fila, columna] -> elemento en la fila ‘fila’ y la columna ‘columna’
* df.loc[filas, columnas] -> dataframe con los elementos en las filas con los nombres de la lista ‘filas’ y las columnas con los nombres de la lista ‘columnas’
* df.loc[columna] -> serie con los elementos de la columna ‘columna’
* df.columna -> serie con los elementos de la columna ‘columna’, sólo funciona cuando el nombre de la columna no tiene espacios en blanco

Creación desde un archivo de datos

df3=pd.read\_csv('./datasets/archivo.csv')

df3.head()

df3.to\_csv('./datasets/archivo\_output.csv',index=False)

# Visualización

Repaso

* df.copy() -> copia del dataset para no alterar el original
* df.shape -> tupla con la cantidad de filas y columnas
* df.columns.tolist() -> lista de nombres de las columnas
* df.rename -> cambiar nombre a una columna
* df.isna() -> lista la cantidad de datos nulos por columna
* df.count() -> lista la cantidad de datos no nulos por columna
* pd.cosstab(df[columna1], df[columna2])
* df.groupby([columna1, columna2])[columna3].sum()
* df.describe() -> count, mean, std, min, percentages, max
* df.drop(columna, axis=’columns’, inplace=True) -> elimina columna
* df.drop(df[df[columna]==indice].index, axis=’index’. inplace=True) -> elimina fila por condición
* df.loc[df.loc[:,columna]==valor,columna]=otro\_valor -> cambia valores de features
* df.loc[:,columna].replace(to\_replace=[un\_valor,otro\_valor],value=[algun\_valor,algun\_otro\_valor],inplace=True) -> cambia valores de features

Gráficos

* De barras
* Histogramas
* De densidad
* Boxplots
* De a pares
* Dispersogramas