https://pythondiario.com/2019/02/pandas-en-python-data-science.html

Pandas en Python - Data Science

PythonDiario febrero 1, 2019

Pandas es sin duda el paquete más importante de **Python** utilizado para **Data Science**. No solo ofrece muchos **métodos** y **funciones** que facilitan el trabajo con los datos, sino que se ha optimizado para la velocidad, lo que le brinda una ventaja significativa en comparación al trabajo con datos numéricos utilizando las funciones integradas en **Python**.

Es común que al comenzar con **Pandas** tengamos problemas para recordar todas las **funciones** y **métodos** que necesitemos, a veces es bueno tener una referencia útil, por eso esta entrada pretende reunir las funciones y métodos más utilizadas.

Puedes ver también la entrada: **Introducción a Pandas para el análisis de datos**

Para este listado se utiliza la abreviación:

df - Cualquier Objeto DataFrame Pandas
s - Cualquier Objeto de Serie Pandas

Para comenzar es necesario la importación:

```
import pandas as pd
import numpy as np
```

Importador de Datos

pd.read_csv(filename) - De un archivo CSV
pd.read_table(filename) - Desde un archivo de texto delimitado (como TSV)
pd.read_excel(filename) - De un archivo Excel
pd.read_sql(query, connection_object) - Lee desde una BaseDeDatos/Tabla SQL
pd.read_json(json_string) - Lee desde una cadena, URL o archivo con formato JSON
pd.read_html(url) - Analiza una URL html, una cadena o un archivo y extrae tablas a una lista
pd.read_clipboard() - Toma el contenido del porta papeles
pd.DataFrame(dict) - Desde un diccionario

Exportador de Datos

df.to_csv(filename) - Escribir en un archivo CSV
df.to_excel(filename) - Escribir en un archivo Excel
df.to_sql(table_name, connection_object) - Escribir en una tabla SQL
df.to_json(filename) - Escribir en un archivo con formato JSON

Crear objetos de Test

Útil para probar segmentos de código
pd.DataFrame(np.random.rand(20,5)) - 5 Columnas y 20 filas con Floats aleatorios
pd.Series(my_list) - Crea series de una lista iterativa
df.index = pd.date_range('1900/1/30', periods = df.shape[0]) - Añade un índice de fecha

Visualizar / Inspeccionar Datos

df.head(n) - Primeras n filas del DataFrame
df.tail(n) - Las últimas n filas del DataFrame
df.shape() - Número de filas y columnas
df.info() - Índice, tipo de datos y memoria
df.describe() - Estadísticas resumidas de columnas numéricas
s.value_counts(dropna=False) - Ver valores y recuentos únicos
df.apply(pd.Series.value_counts) - Valores únicos para todas las columnas

Selección

df[col] - Devuelve la columna con la etiqueta col como Serie
df[[col1, col2]] - Devuelve columnas como un nuevo DataFrame
s.iloc[0] - Selección por posición
s.loc['index_one'] - Selección por índice
df.iloc[0,:] - Primera fila
df.iloc[0,0] - Primer elemento de la primera columna

Limpieza de datos

df.columns = ['a', 'b', 'c'] - Renombrar columnas
pd.isnull() - Comprueba valores nulos, devuelve Boolean Arrays
pd.notnull() - El opuesto a pd.isnull()
df.dropna() - Elimina todas las filas que contienen valores nulos
df.dropna(axis=1) - Elimina todas las columnas que contienen valores nulos

df.dropna(axis=1, thresh=n) - Elimina todas las filas que tienen menos de n valores no nulos
df.fillna(x) - Remplaza todos los valores nulos por x
s.fillna(s.mean()) - Remplaza todos los valores nulos por la media
s.astype(float) - Convierte el tipo de datos a float
sreplace(1, 'one') - Remplaza todos los valores iguales a 1 con 'one'

s.repalce([1, 3], ['one', 'three']) - Remplaza todos los 1 por 'one' y 3 por 'three'
df.rename(columns=lambda x: x + 1) - Cambio de nombre de columnas en masa
df.rename(columns={'old_name':'new_name'}) - Renombrar seleccionando columna
df.set_index('column_one') - Cambiar el índice
df.rename(index=lambda x: x + 1) - Cambio el índice en masa

Filtro, orden y agrupamiento

df[df[col]> 0.5] - Filas donde la columna col es mayor que 0,5
df[(df[col] > 0.5]) & (df[col] < 0.7)] - Filas donde 0.7 > col > 0.5

df.sort_values(col1) - Ordenar los valores por la col1 en orden ascendente
df.sort_values(col2, ascending=False) - Ordena los valores por col2 en orden descendente
df.sort_values([col1, col2], ascending=[True, False]) - Ordena los valores por la col1 de forma ascendente y luego por la col2 en orden descendente
df.groupby(col) - Devuelve un objeto groupby para los valores de una columna
df.groupby([col1, col2]) - Devuelve un objeto groupby para valores de múltiples columnas
df.groupby(col1)[col2] - Devuelve la media de los valores en col2, agrupados por los valores en col1 (la media se puede remplazar con casi cualquier función de la sección Estadísticas)
df.pivot_table(index=col1, values=[col2, col3], aggfunc=mean) - Crea una tabla dinámica que agrupa por col1 y calcula la media de col2 y col3

df.groupby(col1).agg(np.mean) - Encuentra promedio en todas las columnas para cada grupo de col1 único
df.apply(np.mean) - Aplica la función np.mean() en cada columna
nf.apply(np.max, axis=1) - Aplica la función np.max() en cada fila

Unir / Combinar

df1.append(df2) - Agrupa las filas en df1 al final de df2 (las columnas deben ser idénticas)
pd.concat([df1, df2], axis=1) - Agrega las columnas en df1 al final de df2 (las filas deben ser idénticas)

df1.join(df2, on=col1, how='inner') - Une las columnas en df1 con las columnas en df2 donde las filas para col tienen valores idénticos. También puede utilizarse: left, right, outer, inner.

Estadísticas: Todas estas funciones también se pueden aplicar a una serie

df.describe - Resumen de estadísticas para columnas numéricas
df.mean() - Devuelve la media de todas las columnas
df.corr() - Devuelve la correlación entre columnas en un DataFrame
df.count() - Devuelve el número de valores no nulos en cada columna DataFrame
df.max() - Devuelve el valor más alto en cada columna
df.min() - Devuelve el valor más bajo en cada columna
df.median() - Devuelve la media de cada columna
df.std() - Devuelve la desviación estándar de cada columna

Y por aquí la entrada de hoy. Quiero destacar que estos datos (**Pandas en Python**) fueron sacados de la fuente: dataquest.io

Espero pueda ser de ayuda. Saludos