dvva18 pandas Cheat Sheet

by dfadda via cheatography.com/59692/cs/15632/

Importare pandas e numpy

`import pandas as pd`
importa la libreria pandas

`import numpy as np`
importa la libreria numpy

`pd.set _op tio n(' max _co lumns', 300)`
limitare il numero di colonne stampate in output:*

`pd.set _op tio n(' max _rows', 1000)`
limitare il numero di righe stampate in output:*

Creare una serie

`my_data = [10,20,30]` e `labels = ['A','B','C']`
`pd.Ser ies (da ta= my_ data, index= labels)`

Caricare un file in un dataframe

`df=pd.r ea d_c sv( " fil e.c sv")`
Legge un file csv

parametri aggiuntivi:

`header` specifca la riga da usare come header (int type), nel caso non sia presente: `header =None`

`sep=';'` specifica il separatore usato nel file csv

`names= ['c ol1 ',' col2']` esplicita il nome delle colonne del dataframe, utile soprattutto in caso non ci sia header nel file

`parse_ dates = ['col1 ',' col2']`
lista di interi nomi di colonne da tentare di parsare come data.

`df=pd.r ea d_e xce l("n ome fil e.x lsx ")` legge un file excel

Salvare un dataframe in csv

`df.to_ csv ('p ath /fi le_ nam e.c sv" ,se p=';', index= False)`
salva il dataframe in un file csv (rimuovendo l'indice e usando come sepraratore il ;)

Esplorare un dataframe

`df.shape`
restituisce (numero righe, numero colonne)

`df[df.i sn ull ().a ny (ax is=1)]`
controlla se ci sono valori nulli

`df.dtypes`
restituisce il tipo di variabile di ogni colonna

`df.head()`
restituisce le prime (n) righe di un dataframe (default = 5)

`df.tail()`
restituisce le ultime (n) righe di un dataframe (default = 5)

`df.columns`
restituisce l'elenco delle colonne del dataframe

`df.des cribe()`
restituisce una descrizione del dataframe con le statistiche di base

Gestire i dati mancanti

`df.dro pna (ax is=0)`
Elimina le righe con valori null presenti in qualsiasi colonna

`df.dro pna (ax is=1)`
Elimina le colonne con valori null

`df.dro pna (ax is=0, thresh=5)`
Elimina le righe con almeno 5 valori null presenti in qualsiasi colonna

`df.fil lna (va lue ='V ALORE NUOVO')`
Riempi le righe con valori null usando il valore *value* indicato tra parentesi

Reshape dei dati

`df.sor t_v alu es( by= ['c ol1 ',' col2'])`
ordina il dataframe per colonna o array di colonne

`df.sor t_v alu es( by= 'co l1' ,as cending= False)`
ordina il dataframe per colonna in ordine discendente

Reshape dei dati (cont)

`df.sor t_i ndex()`
ordina l'indice del dataframe

`df.res et_ index()`
resetta l'indice del dataframe usando l'ordine c
colonna

`df.res et_ ind ex( nam e='new name`
assegna il nome alla colonna indice durante la

`df.dro p(c olu mns =[' col 1', 'co`
elimina le colonne 'col1' e 'col2'

`df.ren ame (co lum ns= {'o ld_ col umn2'})`
rinomina le colonne (usando un dizionario)

`df.con cat ([d f1, df2])`
aggiunge le righe del df2 al df1

`df.con cat ([d f1, df2], axis=1)`
aggiunge le colonne del df2 al df1

ricavare un sottoinsieme di un dataframe

per righe

`df[df[ 'co lon na1 ']>n]`
seleziona tutte le righe dove nella *colonna1* i v
maggiori di *n*

`df.dro p_d upl ica tes()`
rimuove le righe duplicate

`df.ilo c[1 0:20]`
seleziona tutte le righe dalla 10 alla 20

`df[df[ 'co lum n'].is in( ['v alu alu e2'])]`
seleziona tutte le righe dove il valore nella colo
column è uguale a *value1* o *value2*

`df['co lum n'].un ique()`
restituisce i valori unici di una colonna

per colonne

`df['co lon na1']` o `df.col onna1`
seleziona la colonna 1

ricavare un sottoinsieme di un dataframe (cont)

```
df[['c olo nna 1', 'co lon na2 ',' colonn a4']]
```
Seleziona le colonne 1,2,4

```
df.ilo c[1 0:20, [1,2,5]]
```
seleziona tutte le righe dalla 10 alla 20 e le colonne in posizione 1,2,5

```
df.loc[:, 'x2':'x4']
```
seleziona tutte le colonne dalla posizione x2 alla posizione x4 (compresa)

```
df.loc [df ['a ']> 10, ['a ','c']]
```
seleziona le colonne *'a'* e *'c'* dove il valore di a è maggiore di 10

Varie

```
df.columns
```
restituisce l'elenco delle colonne

```
del df['co l_1']
```
elimina una colonna

Riassumere i dati

```
len(df)
```
number of rows in a dataframe

```
df['co lum n1' ].n uni que()
```
numero di valori unici in una colonna

```
df['co lum n1' ].sum()
```
somma dei valori di una colonna

```
df['co lum n1' ].min()
```
minimo dei valori di una colonna

```
df['co lum n1' ].max()
```
massimo dei valori di una colonna

```
df['co lum n1' ].m ean()
```
media dei valori di una colonna

```
df['co lum n1' ].var()
```
varianza dei valori di una colonna

```
df['co lum n1' ].std()
```
deviazione standard dei valori di una colonna

Riassumere i dati (cont)

```
df['co lum n1' ].c ount()
```
count dei valori non nulli di una colonna

```
df['co lum n1' ].m edian()
```
mediana dei valori di una colonna

```
df['co lum n1' ].q uan til e([.25 ,.75])
```
quantile dei valori di una colonna

```
df['co lum n1' ].v alu e_c ount()
```
count di ogni valore di una colonna

Gestione oggetti di tipo datetime

```
df['Da ta' ]=p d.t o_d ate tim e(d f[' Data'])
```
converte stringhe in oggetti datetime

```
df['Da ta' ].d t.r oun d('1s')
```
arrotonda la data al secondo

```
df['Du rat ion ']= (df ['D ata 2'] -df ['D ata 1'] ).d t.s econds
```
Calcola la differenza tra due date restituendo un risultato in seocndi

```
df['Da ta' ].d t.date
```
estrae solo la data da data ora

```
df['Da ta' ].d t.month
```
estrae il mese

```
df['Da ta' ].d t.week
```
estrae la settimana dell'anno

```
df['Da ta' ].d t.w eekday
```
estrae il giorno della settimana

```
df['Da ta' ].d t.w eek day _name
```
estrae il giorno della settimana (come nome)

```
df['Da ta' ].d t.time
```
estrae l'ora completa

```
df['Da ta' ].d t.hour
```
estrae l'ora

```
df['St agi one ']= pd.cut(     (df['D ata '].d t.d ay ofyear + 11) % 36 autunno'] )
```
crea una colonna con la corrispondente stagione

Crea ed elimina colonne/righe

```
df['ne w_c olumn'] = df['co l_1'] _2']
```
crea una nuova colonna somma di due colonn

```
df = df.dro p(' col_1', axis=1)
```
elimina la colonna *col_1*

```
df = df.drop(0, axis=0)
```
elimina la riga con indice *0*