

Esercizi lezione 6

Esercizio

Abbiamo una lista di liste:

```
mat = [[0, 1, 2, 3, 4],  
       [5, 6, 7, 8, 9]  
       [10, 11, 12, 13, 14]]
```

Che tipo di struttura dati o matematica potrebbe rappresentare? Notare che tutte le liste "interne" sono della stessa dimensione

Come facciamo per accedere ad un elemento in particolare?

Esercizio

Trasformiamo la lista dell'esercizio precedente

```
mat = [[0, 1, 2, 3, 4],  
       [5, 6, 7, 8, 9]  
       [10, 11, 12, 13, 14]]
```

in un array NumPy:

```
mat = np.array(mat)
```

Come facciamo per accedere ai singoli elementi?

Esercizio

Abbiamo il seguente array NumPy:

```
linear_data = np.array([x for x in range(27)])
```

Lo ridimensioniamo mediante il metodo `.reshape()`:

```
reshaped_data = linear_data.reshape((3, 3, 3))
```

Quante dimensioni ha il nuovo array?

Come facciamo per accedere ai singoli elementi?

Esercizio 1/3

Abbiamo un territorio cittadino diviso in quattro quadranti; in ognuno di essi c'è uno store:

| | |
|-----------|-----------|
| UL | UR |
| LL | LR |

Vogliamo contare quanti clienti vengono serviti dai diversi store: creiamo una struttura dati adeguata con un array NumPy, dove il conteggio di ogni store parta da zero.

Esercizio 2/3

Creiamo un ciclo di 100 iterazioni che riempia, a caso, uno dei quattro quadranti, simulando quindi diversi clienti serviti dai vari store.

Alla fine, che valore ha lo store UR? E lo store LL?

Quali store hanno avuto più di 25 clienti?

Quanti clienti hanno avuto di preciso questi store?

Qual è la loro somma?

Esercizio 3/3

Ripetiamo la traccia precedente, ma stavolta abbiamo nove quadranti e 200 iterazioni:

| | | |
|-----------|-----------|-----------|
| UL | UC | UR |
| ML | MC | MR |
| LL | LC | LR |

L'anno prossimo ci aspettiamo 4 volte tanto i valori appena ottenuti; simuliamo velocemente il tutto e stampiamo il risultato.

Esercizio

Chiediamo all'utente di inserire un numero, dividiamolo per 5 e poi stampiamo il risultato a video.

Gestiamo correttamente le eccezioni nel caso l'utente immetta un input errato.

Esercizio

In una catena di montaggio abbiamo una struttura metallica di 28.75 cm di lunghezza; per assicurarne la stabilità, è necessario inserire 15 rivetti, dei quali uno all'inizio e uno alla fine, e tutti quanti separati dalla stessa distanza; come possiamo calcolare i punti esatti in cui inserire i rivetti tramite NumPy?

Esercizio 1/2

Scarichiamo l'Iris dataset da qui: <https://archive.ics.uci.edu/dataset/53/iris>

Troveremo un file `.data`, che è un CSV, e un file `.names` con i metadati; questa versione del dataset non ha i nomi di colonna.

- Leggiamo il file e carichiamolo in un `DataFrame` mediante `pd.read_csv()` senza utilizzare altri parametri
- Stampiamo le prime cinque righe
- Stampiamo i nomi di colonna: sono corretti?

Esercizio 2/2

Dal dataset di prima:

- Leggiamo il file e carichiamolo in un `DataFrame`, aggiungendo i nomi di colonna — che si trovano nel file `.names` — come parametro di `pd.read_csv()`
- Stampiamo le prime cinque righe e le ultime dieci
- Stampiamo un riepilogo dei descrittori statistici del dataset

Esercizio

Andiamo a questo link e scarichiamo una serie di dataset:

<https://www.kaggle.com/datasets/ahmettezcantekin/beginner-datasets>

Tra i vari dataset presenti, ce n'è uno che contiene diverse qualità di vini e le misure di diverse proprietà organolettiche, `wine.csv`;

Leggiamo quindi il dataset `wine.csv`, e visualizziamone le prime e le ultime righe;

Leggiamo nuovamente il dataset, ma stavolta considerando soltanto le ultime 6 colonne, ricordandoci che in totale il dataset ha 13 colonne;

Visualizziamo un resoconto dei descrittori statistici di questa versione.

Esercizio

Consideriamo il seguente dizionario:

```
fatturati_dict = {1997: 12_000, 1998: 15_000,  
                  1999: 20_000, 2000: 23_000, 2001: 25_000,  
                  2002: 17_000, 2003: 14_000, 2004: 21_000}
```

Consideriamo ora la seguente Series:

```
fatturati_series = pd.Series([12_000, 15_000, 20_000,  
                              23_000, 25_000, 17_000, 14_000, 21_000],  
                              index=range(1997, 2005))
```

Possiamo accedere alle stesse informazioni nello stesso modo:

```
fatturati_dict[1997]  
fatturati_series[1997]
```

Dunque qual è la differenza tra i due tipi di dato?

Cosa potremmo fare con la `Series` che non possiamo fare con il dizionario?

Esercizio

Abbiamo la seguente matrice:

| | | | |
|----|----|---|----|
| 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 |
| 20 | -4 | 0 | 42 |

Creiamo un `ndarray` con gli stessi valori.

Ci sono due modalità: inizializzare un array e poi inserire i valori nelle posizioni adatte, oppure creare una lista di liste e poi effettuare un casting.

Esercizio

Creiamo il seguente ndarray 5×5 :

| | | | | |
|----|----|----|----|----|
| 10 | 22 | 21 | 8 | 9 |
| 9 | 42 | 3 | 18 | 11 |
| 5 | 4 | 30 | 12 | 29 |
| 37 | 31 | 7 | 2 | 26 |
| 8 | 6 | 4 | 33 | 15 |

Per ogni valore, sottraiamo il minimo (2) e poi dividiamo il risultato per il massimo (42) meno il minimo.



GRAZIE
Epicode