fondo
sociale europeo

Elementi di Programmazione con Python e Analisi dei Dati
Lezione 5: pulizia, preparazione e sistemazione dei dati

Stefano Andreozzi, PhD

Formazione continua individuale – Id Attività: 2530775 Codice Corso: B341-1-2019-0

7 dicembre 2020

per una crescita intelligente,
sostenibile ed inclusiva

www.regione.piemonte.it/europa2020

INIZIATIVA CO-FINANZIATA CON FSE

Piano di battaglia

- Piccolo intermezzo su strutture dati e iterazioni
- McKinney2017
  - Ch. 7: Data Cleaning and Preparation
  - Ch. 8: Data Wrangling: Join, Combine, and Reshape
- Espressioni regolari: https://www.python.it/doc/howto/Regex/regex-it/regex-it.html

Strutture dati *multidimensionali*

- `[]`: usato per definire tipi di dato mutabili (liste) e per indicizzazione
- `np.array([])`: costruttore del tipo dato array di numpy
- `()`: definizione di tuple
- `{}`: dizionari e set

Quando si cicla nei **dizionari**, la chiave e il valore corrispondente possono essere recuperati contemporaneamente usando il metodo `items()`

```
1 rediroma = {'tito': 'tazio', 'numa': 'pompilio', 'tullo': 'ostilio', 'anco': 'marzio', '
      tarquinio': 'prisco', 'servio': 'tullio', 'tarquinio': 'il superbo'}
2 for r, v in rediroma.items():
3   print(r, v)
```

L'output è come ce lo aspettiamo? e perché?

Quando si cicla nelle **liste**, l'indice della posizione e il valore corrispondente possono essere recuperati contemporaneamente usando il metodo `enumerate()`

```
1 for i, v in enumerate(['tic', 'tac', 'toe']):
2   print(i, v)
```

`zip()` associa gli elementi degli oggetti iterabili in una tupla.

L'argomento della funzione è uno o più oggetti iterabili (stringhe, liste, tuple, ecc.) separati da virgola.

La funzione restituisce in output una tupla con gli elementi degli oggetti secondo l'ordine di posizione.

Esempio: definisco una stringa e una lista, rispettivamente nelle variabili `s` e `l`, poi ciclo sulla loro versione *zippata*:

```
s="ABCD"
l=["Alfa", "Beta", "Gamma", "Delta"]
for i in zip(s,l):
  print(i)
```

Per ciclare all'incontrario, prima si specifica la seguenza nel senso normale, poi si invoca la funzione `reversed()`:

```
for i in reversed(range(1, 10, 2)):
  print(i)
```

Per ciclare secondo un ordinamento, si invoca la funzione `sorted()`. È un metodo `inplace`?

```
gloria = ['ho','lasciato','la','patente','sul','tavolo','accanto','alla','frutta']
for i in sorted(gloria):
  print(i)
```

Usando `set()` si eliminano gli elementi duplicati. L'uso di `sorted()` in combinazione con `set()` è un modo per ciclare sugli elementi unici di una lista messi in ordine alfabetico.

```
cesto = ['mela', 'arancia', 'mela', 'pera', 'arancia', 'banana']
for f in sorted(set(cesto)):
  print(f)
```

Consiglio: **mai modificare una lista quando si cicla su di essa**. Crearne una nuova:

```
import math
raw_data = [56.2, float('NaN'), 51.7, 55.3, 52.5, float('NaN'), 47.8]
filtered_data = []
for value in raw_data:
  if not math.isnan(value):
    filtered_data.append(value)
```

Dizionari, liste, tuple e stringhe sono tutti oggetti iterabili, ovvero dei contenitori da cui ottenere l'iteratore tramite il metodo `iter()` e il valore successivo invocando `next()` sull'iteratore:

```
1 gloria = ['ho','lasciato','la','patente','sul','tavolo','accanto','alla','frutta']
2 iteratore = iter(gloria)
3 print(next(iteratore))
```

Se vado oltre l'ultimo oggetto della lista ottengo una eccezione del tipo `StopIteration`.

- Pipeline: **O**btain; **S**crub / Clean; **E**xplore / Visualize; **M**odel; I**N**terpret.
- The main steps in **data wrangling** (*sistemazione*) are as follows:
  - **Discovering**: The first step of data wrangling is to gain a better understanding of the data, different data is worked and organized in different ways.
  - **Structuring**: The next step is to organize the data. Raw data is typically unorganized and much of it may not be useful for the end product. This step is important for easier computation and analysis in the later steps.
  - **Cleaning**: There are many different forms of cleaning data, for example one form of cleaning data is catching dates formatted in a different way and another form is removing outliers that will skew results and also formatting null values. This step is important in assuring the overall quality of the data.
  - **Enriching**: At this step determine whether or not additional data would benefit the data set that could be easily added.
  - **Validating**: This step is similar to structuring and cleaning. Use repetitive sequences of validation rules to assure data consistency as well as quality and security. An example of a validation rule is confirming the accuracy of fields via cross checking data.
  - **Publishing**: Prepare the data set for use downstream, which could include use for users or software. Be sure to document any steps and logic during wrangling.
- Questi step costituiscono un processo iterativo che deve portare ad un set di dati pulito e usabile per le analisi. Si tratta di un processo noioso ma gratificante.