



fondo
sociale europeo

Elementi di Programmazione con Python e Analisi dei Dati

Lezione 1: Introduzione

Stefano Andreozzi, PhD

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

23 novembre 2020



REGIONE
PIEMONTE

per una crescita intelligente,
sostenibile ed inclusiva

www.regione.piemonte.it/europa2020

INIZIATIVA CO-FINANZIATA CON FSE

- prerequisiti: programmazione imperativa, matematica del liceo
- lunedì e mercoledì sera: 18h30 – 21h30
- 1 h accoglienza, 32 h di lezione
- se fosse un corso universitario, sarebbero 4 CFU (1 CFU = 25 h, di cui 8 di lezione)
 - gli allievi dovrebbero aggiungere ulteriori **68 h di studio individuale** 😊
- vacanze natalizie: **no lezione** 21, 23, 28, 30 dicembre, 4, 6 gennaio
- mercoledì 20 gennaio 2021: **1 h ripasso, 1 h verifica**

Ogni lezione:

- appello
- 1 h 15' circa teoria, 10' circa pausa, 1 h 30' circa “hands on”
- materiale didattico fornito dal docente + rimandi per approfondimenti
- “compiti per casa” (non obbligatori ma fortemente consigliati)

- laurea in Controlli Automatici, Dottorato in Chimica e Ingegneria Chimica
- lavoro in un'azienda della provincia di Torino, mantenendo una posizione accademica “onorifica”
- 15+ anni “esperienza” nel mondo dell’ICT
- ho lavorato su parecchie cose differenti, dal diabete alla logistica di magazzino, dai treni ad alta velocità alle scommesse sportive, dai biocarburanti all’epigenetica dei gemelli
- non sono “esperto” di nessuna di queste cose in particolare
- grandi passioni: analisi numerica, statistica computazionale
- **sicuramente la maggior parte di voi sa programmare meglio di me in molti contesti**
- contatti:
 - aggiungetemi su LinkedIn se vi va
 - email: stefano.andreozzi@gmail.com
 - WhatsApp: +39 328 64 89 157
 - **usare in caso di stretta necessità, non abusare!**
- **se volete farmi vedere qualsiasi cosa sul vostro schermo: per favore, ingranditelo**
- vogliamo usare un forum o un gruppo WhatsApp o similare per discussioni e supporto fuori dalle lezioni?

- un fatto, ossia un concetto che può essere comunicato, interpretato o elaborato da esseri umani o da strumenti automatici
 - una parola che denota un oggetto (es. "chiave", "auto", ecc...)
 - una cifra numerica
 - un cartello stradale
 - ...
- **un'informazione per essere utilizzabile deve essere interpretabile in modo univoco**
- dato: ogni rappresentazione dell'informazione mediante opportuni simboli dell'alfabeto ("30")
- informazione = dato + descrittore
 - dati utilizzabili solo se vengono qualificati, usando un descrittore ("kg")
 - descrittore: entità che qualifica e che consente di interpretare in modo corretto il dato
 - "30 kg"

Peter Sondergaard, Vice Presidente Esecutivo di Gartner, azienda leader mondiale nella consulenza strategica (2011):

Information is the oil of the 21st century, and Analytics is the combustion engine

Harvard Business Review, ottobre 2012:

Data Scientist: The Sexiest Job of the 21st Century

- Pipeline: **O**btain; **S**crub / Clean; **E**xplore / Visualize; **M**odel; **I**Nterpret.
- acquisizione, pulizia, trasformazione, modellazione, visualizzazione e interpretazione
- atto ad aumentare la conoscenza di informazioni utili alle decisioni aziendali

analisi diagnostica: perché è successa una certa cosa

analisi predittiva: cosa succederà, verosimilmente

analisi prescrittiva: quali azioni intraprendere (sulla base di analisi diagnostica e predittiva)

Basi di dati memorizzazione e integrazione delle informazioni

- relazionali
- non relazionali (big data: massa di dati non strutturati – testo, audio, video – prodotti e catturati a frequenza elevata)

Analisi dei testi ed elaborazione del linguaggio naturale analisi quantitative su testo (*sentiment analysis*)

Data mining ricerca di schemi coerenti e relazioni fra variabili

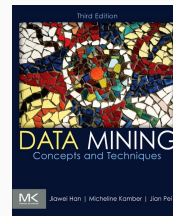
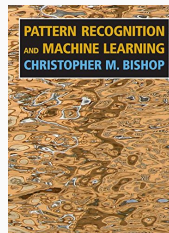
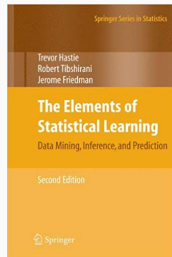
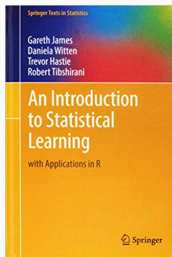
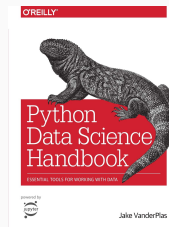
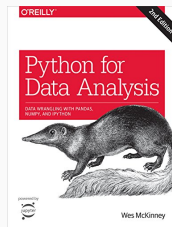
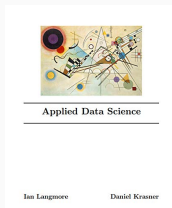
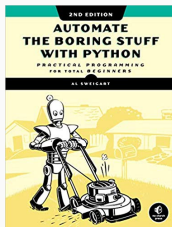
Elaborazione di serie temporali elaborazione di segnali numerici nel dominio del tempo e della frequenza

Analisi di reti complesse collezioni di entità arbitrarie interconnesse

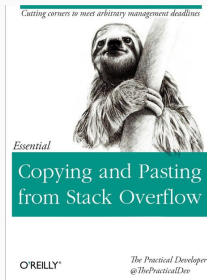
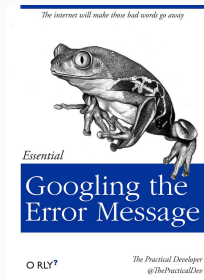
- Statistica**
- descrittiva (media, deviazione standard, correlazione, regressione lineare, MLE, ecc....)
 - inferenziale (da un sottoinsieme dei dati trovare conclusioni su tutto l'insieme, campionamento)
 - predittiva (machine learning)

Analisi prescrittiva ottimizzazione

Rappresentazione dei dati tramite tabelle o grafici



- **la disponibilità di materiali/libri sulla materia sorpassa la capacità umana di fruirne nella sua totalità**
- in pratica, nella maggiore totalità dei casi:



- alcuni riferimenti:
 - W. McKinney, *Python for Data Analysis*, 2nd edition, O'Reilly, 2017 ([info](#) | [codice](#))
 - J. Grus, *Data Science from Scratch*, O'Reilly, 2015 ([info](#) | [codice](#))
 - A. B. Downey, *Think Stats*, 2nd edition, Green Tea Press, 2014 ([scarica](#) | [info](#) | [codice](#))

- settore affetto da rapida obsolescenza (i linguaggi passano...)
- programmatori come “blue collar workers” del 21esimo secolo
- l’automazione cancellerà in futuro molti lavori in ambito IT, specialmente nei compiti ripetitivi

Ma

- **il contributo umano rimarrà essenziale nell’Analisi dei Dati**
- **i concetti, che sono intrinsecamente logici e matematici, rimangono**

Quindi:

- maggiore focus su aspetti teorici piuttosto che tecnologico-implementativi



**PER PROGRAMMARE BENE BISOGNA ESSERE DISPOSTI A
IMPARARE BENE UN PO’ DI MATEMATICA E DI STATISTICA!**

- un foglio Excel usato come un database




Health policy

Covid: how Excel may have caused loss of 16,000 test results in England

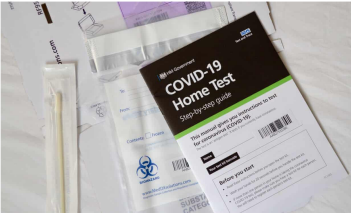
Public Health England data error blamed on limitations of Microsoft spreadsheet

- [Coronavirus - latest updates](#)
- [See all our coronavirus coverage](#)

Alex Hern UK technology editor
@alexhern
Tue 6 Oct 2020 08.21 BST



2,881



▲ More than 50,000 potentially infectious people may have been missed by contact tracers after 15,841 positive tests were left off the daily figures. Photograph: Simon Leigh/Alamy

A million-row limit on Microsoft's Excel spreadsheet software may have led to Public Health England misplacing nearly 16,000 Covid test results, it is understood.

The data error, which led to **15,841 positive tests being left off the official daily figures**, means than 50,000 potentially infectious people may have been missed by contact tracers and not told to self-isolate.

- Guido van Rossum: sviluppato alla fine del 1980 e pubblicato nel 1991. Gestione delle eccezioni, funzioni e classi con ereditarietà.
- 1994: newsgroup Usenet `comp.lang.python`
- Usato da grandi realtà come United Space Alliance (principale committente del supporto allo shuttle) e Industrial Light & Magic (studio di animazione e VFX della Lucasfilm)
- **Python 2** (fine 2000): implementazione della PEP (Python Enhancement Proposal), una specifica tecnica che
 - fornisce delle informazioni ai membri della comunità Python
 - descrive una nuova funzionalità del linguaggio
- garbage collector
- supporto Unicode
- unificazione dei tipi e delle classi in una gerarchia
- Versione 2.7.17: penultima release, solo correzioni di bug, **cesserà completamente quest'anno**

- Python 3 (fine 2008): correzione di difetti di progettazione intrinseci delle precedenti versioni del linguaggio.
 - istruzione print
 - divisione fra interi
 - supporto a Unicode.
- non retrocompatibile con Python 2
- **versione di riferimento del corso: 3.8.2, ambiente Windows 10**

Per installare JupyterLab

```
1 pip install jupyterlab
```

Per eseguire JupyterLab

```
1 jupyter lab
```

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the best programming language or the language in which most lines of code have been written.

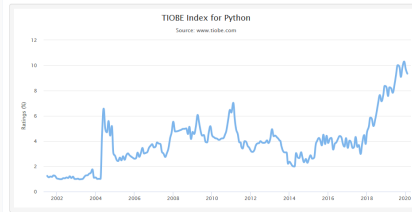
The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](https://www.tiobe.com/tiobe/index.php?id=1&lang=en).

Feb 2020	Feb 2019	Change	Programming Language	Ratings	Change
1	1		Java	17.358%	+1.48%
2	2		C	16.766%	+4.34%
3	3		Python	9.345%	+1.77%
4	4		C++	6.164%	-1.28%
5	7	▲	C#	5.927%	+3.08%
6	5	▼	Visual Basic .NET	5.862%	-1.23%

▲ Highest Position (since 2001): #3 in Feb 2020

▼ Lowest Position (since 2001): #19 in Feb 2003

🏆 Language of the Year: 2007, 2010, 2018



- linguaggio di programmazione general-purpose molto diffuso
- molto espressivo
- facilità di scrittura di programmi
- facilità nel leggere i dati
- pacchetti: software pronto all'uso, aggiornati e documentati

- scalabilità e rapidità di prototipazione
- visualizzazione e grafica
- vasta knowledge base
- sequenza di istruzioni
 - non interattiva: su file di script, uso di ambiente di sviluppo
 - interattiva: prompt dei comandi

C

- **Basso livello:** allocazione esplicita della memoria (gestione dei puntatori)
- **Compilato:** file .c, gcc, codice macchina
- **Fortemente tipizzato:** il tipo della variabile va dichiarato esplicitamente
- **Procedurale**

In Python:

- Le variabili non vanno dichiarate prima di essere usate
- Non è necessario usare i punti e virgola alla fine di ogni istruzione
- I costrutti (if-else, for, funzioni) usano i due punti al posto delle graffe di apertura
- devono essere correttamente indentati con spazi o tabulazione (evitare di usarli entrambi)
- I commenti si inseriscono con # (singola linea) o `"""commento"""` (multi linea/docstring)

Python

- **Alto livello:** allocazione implicita della memoria
- **Interpretato:** file .py, interprete Python, codice interpretato
- **Debolmente tipizzato:** nessuna dichiarazione esplicita di tipo
- **Procedurale, ad oggetti, funzionale, vettoriale**

- Ci viene insegnato a scrivere codice ma non a prendersene cura.
- *Scrivi il tuo codice come se la prossima persona a doversene occupare fosse uno psicopatico che sa dove abiti.*
- Lo psicopatico posso essere “io stesso fra 6 mesi”
- **correttezza:** essere sicuri che il vostro codice faccia esattamente quello che pensate; qualsiasi modifica facciate, volete essere certi di non aver introdotto errori. Nel caso avvenisse, volete poter tornare indietro.
- **ripetibilità I:** volete essere in grado di ripetere un’analisi ottenendo gli stessi risultati a distanza di tempi anche lunghi. Questo vuol dire tenere traccia di quali siano i requisiti del vostro software, di come si usa, su quali dati e con quali parametri.
- **ripetibilità II:** permettere a qualcun altro di fare lo stesso, possibilmente senza che voi siate lì fisicamente presenti a spiegare passo per passo.
- **riproducibilità:** permettere ad altri di fare lo stesso, e di testare il vostro codice (ed in generale le vostre idee) su altri dati ed altri casi rispetto a quelli da voi esaminati.
- **auditing:** mantenere la storia del progetto, per sapere cosa è stato fatto, quando e perché. Queste sia per mantenere la comprensione acquisita nel tempo che per permettere a dei revisori esterni di verificare quello che avete fatto.

- controllo di versione
- documentazione
- procedure di test
- automazione delle procedure
- progettazione della pipeline di lavoro

Tratte da storie vere:

- il programma ha girato per 36 ore e fallisce all'ultimo step. Non ci sono salvataggi intermedi dei dati.
- la Direzione vi chiede di modificare un grafico, ma per disegnare il grafico dovete far girare di nuovo la simulazione di 36 ore per generare i dati
- vi dedicate ad altri progetti per 6-7 mesi, poi quando è il momento di riprendere in mano il codice non ricordate più cosa avevate fatto e cosa era ancora nella lista delle cose da fare
- il vostro programma richiede una serie di step ben precisi per eseguire correttamente. La donna delle pulizie getta il foglietto dove li avete appuntati
- avete tenuto la documentazione del vostro progetto, ma nell'ultimo backup vi siete dimenticati di copiare l'ultima versione e ora avete la documentazione ed il codice che non coincidono
- passate il vostro progetto ad un collega su una penna USB. Dopo mesi il collega si accorge che la penna è rotta e voi avete formattato il computer.
- prendete in mano il codice di qualcuno, e non avete la più pallida idea del perché una linea di codice sia lì, ma non potete cambiarla perché non capite se è essenziale per il codice.

- editare fogli di calcolo per “pulirli”
 - rimuovere outliers
 - controllo qualità dei dati
 - validazione
- editare tabelle e figure (es. arrotondamenti, formattazioni)
- scaricare dati da un sito web (cliccare collegamenti da un browser)
- copiare/incollare dati da una posizione a un'altra; dividere e cambiare formato ai file di dati
- non esistono cose che si fanno una volta sola, una volta per tutte
- se non possibile altrimenti, le cose fatte a mano devono essere documentate in maniera precisa (è più difficile di quanto sembri)

- se c'è qualcosa che deve essere fatto come parte della propria analisi dei dati, provare a inserirlo nel programma di analisi dei dati (anche se è qualcosa che va fatta una volta sola)
- dare le istruzioni al computer è anche un modo per chiarirsi esattamente cosa va fatto e come
 - usare VCS (*non coperto dal corso*)
 - tenere traccia dell'ambiente in cui sta girando il programma (es. `sinfo()`)
 - usare **ambienti virtuali** (*non coperto dal corso*)
 - se si usa randomizzazione, impostare il seme del generatore pseudo-casuale
- inserire tutte queste operazioni nei propri programmi garantisce la **ripetitività/riproducibilità**