

July 27, 2024

1 File.CSV

Per l'analisi dati dobbiamo imparare a utilizzare file.csv che possiamo reperire in rete, avere da file Excel etc.

Seguiremo due percorsi: con il primo impareremo a caricare file che abbiamo sul nostro pc; con il secondo a far riferimento a pagine web in cui abbiamo reperito i dati.

Esempi di file.csv

<http://www.datiopen.it/>

<https://data.europa.eu/euodp/it/home>

<http://www.protezionecivile.gov.it/attivita-rischi/rischio-sanitario/emergenze/coronavirus> (mappa situazione-CSV-RAW)

```
[ ]: import pandas as pd
      from google.colab import files
      uploaded = files.upload()
```

```
[ ]: import io
      df2 = pd.read_csv(io.BytesIO(uploaded['president_heights.csv']))
```

```
[ ]: print(df2.head(10))
```

| | order | name | height(cm) |
|---|-------|------------------------|------------|
| 0 | 1 | George Washington | 189 |
| 1 | 2 | John Adams | 170 |
| 2 | 3 | Thomas Jefferson | 189 |
| 3 | 4 | James Madison | 163 |
| 4 | 5 | James Monroe | 183 |
| 5 | 6 | John Quincy Adams | 171 |
| 6 | 7 | Andrew Jackson | 185 |
| 7 | 8 | Martin Van Buren | 168 |
| 8 | 9 | William Henry Harrison | 173 |
| 9 | 10 | John Tyler | 183 |

Analisi dati COVID-19

Questo notebook di esempio analizza i dati Italiani dell'epidemia del Covid-19. Il notebook è pensato all'interno della sperimentazione realizzata da:

Claudio Sutrinì (Dipartimento di Fisica, Università di Pavia)

Filippo Pallotta (Dipartimento di Scienze e Alta Tecnologia, Università dell'Insubria, Como)

Davide Passaro (Dipartimento di statistica Università La Sapienza di Roma)

```
[ ]: # https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/  
      ↳ dati-andamento-nazionale/dpc-covid19-ita-andamento-nazionale.csv  
import pandas as pd # libreria pandas per gestire tabelle di dati  
import matplotlib.pyplot as plt #libreria pyplot per fare grafici
```

```
[ ]: # url al file .csv con dati protezione civile  
url = "https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/  
      ↳ dati-andamento-nazionale/dpc-covid19-ita-andamento-nazionale.csv"
```

```
[ ]: # funzione read legge dati dall'indirizzo indicato nell'url  
df = pd.read_csv(url)  
print(df.head(10)) #primi 10 dati  
print("ultimi dati \n",df.tail(5)) # ultimi 5 dati  
print("Stampo nome delle colonne della tabella dei dati importata \n", df.  
      ↳ columns)
```

| | data | stato | ... | casi_testati | note |
|---|---------------------|-------|-----|--------------|------|
| 0 | 2020-02-24T18:00:00 | ITA | ... | NaN | NaN |
| 1 | 2020-02-25T18:00:00 | ITA | ... | NaN | NaN |
| 2 | 2020-02-26T18:00:00 | ITA | ... | NaN | NaN |
| 3 | 2020-02-27T18:00:00 | ITA | ... | NaN | NaN |
| 4 | 2020-02-28T18:00:00 | ITA | ... | NaN | NaN |
| 5 | 2020-02-29T18:00:00 | ITA | ... | NaN | NaN |
| 6 | 2020-03-01T18:00:00 | ITA | ... | NaN | NaN |
| 7 | 2020-03-02T18:00:00 | ITA | ... | NaN | NaN |
| 8 | 2020-03-03T18:00:00 | ITA | ... | NaN | NaN |
| 9 | 2020-03-04T18:00:00 | ITA | ... | NaN | NaN |

[10 rows x 17 columns]

ultimi dati

| | data | stato | ... | casi_testati | note |
|-----|---------------------|-------|-----|--------------|------|
| 227 | 2020-10-08T17:00:00 | ITA | ... | 7365751.0 | NaN |
| 228 | 2020-10-09T17:00:00 | ITA | ... | 7443593.0 | NaN |
| 229 | 2020-10-10T17:00:00 | ITA | ... | 7523702.0 | NaN |
| 230 | 2020-10-11T17:00:00 | ITA | ... | 7592410.0 | NaN |
| 231 | 2020-10-12T17:00:00 | ITA | ... | 7652059.0 | NaN |

[5 rows x 17 columns]

Stampo nome delle colonne della tabella dei dati importata

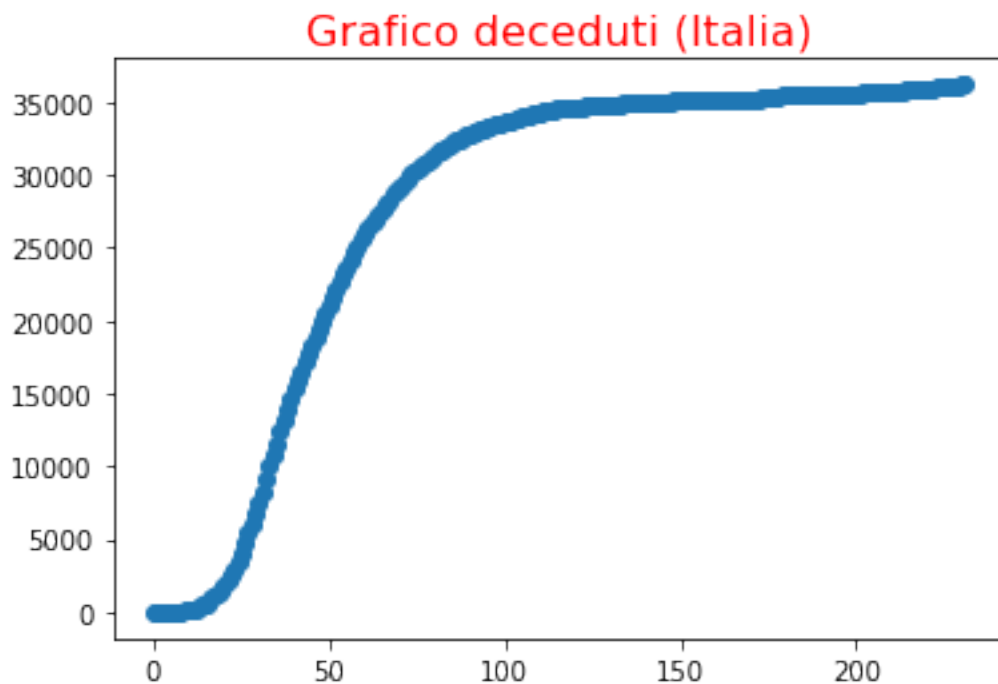
```
Index(['data', 'stato', 'ricoverati_con_sintomi', 'terapia_intensiva',  
      'totale_ospedalizzati', 'isolamento_domiciliare', 'totale_positivi',  
      'variazione_totale_positivi', 'nuovi_positivi', 'dimessi_guariti',  
      'deceduti', 'casi_da_sospetto_diagnostico', 'casi_da_screening',
```

```
'totale_casi', 'tamponi', 'casi_testati', 'note'],
dtype='object')
```

2 Primi grafici

```
[ ]: #grafico singolo

plt.figure(1)
plt.plot(df["deceduti"], 'o-')
plt.title('Grafico deceduti (Italia)', color = "red", fontsize=16)
plt.show()
```



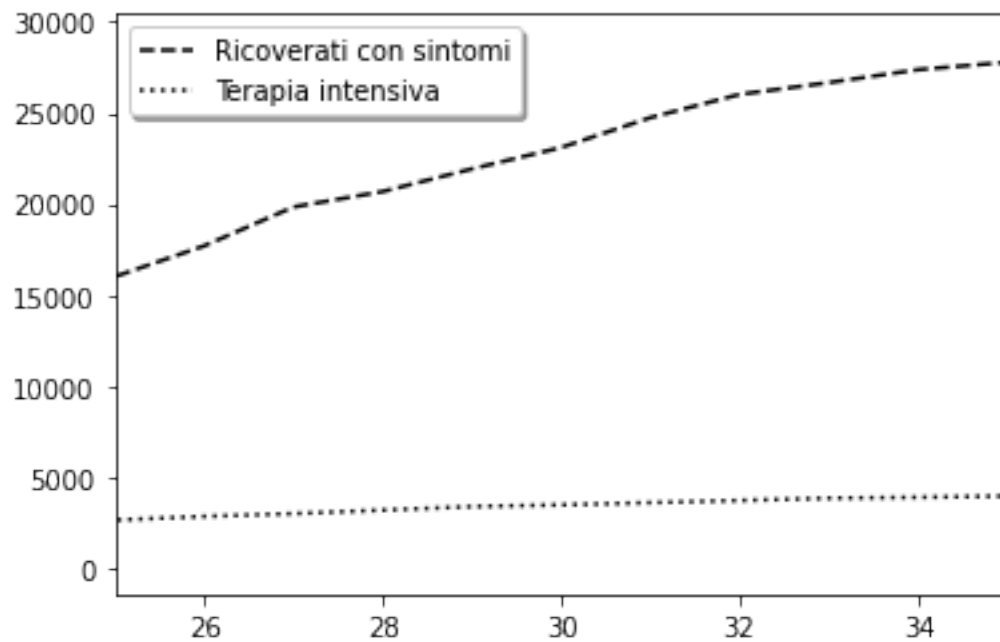
Esercizio: rappresentare in una stessa figura il grafico dei *ricoverati con sintomi* e i *terapia intensiva*. Utilizzare due colori e due stili diversi. Inoltre inserire il titolo e una legenda in alto a sinistra che indichi per ogni grafico a cosa corrisponde.

```
[ ]: # Correzione esercizio
```

Potremmo essere interessati solo ad alcuni giorni

```
[ ]: fig, ax = plt.subplots()
plt.plot(df["ricoverati_con_sintomi"], 'k--', label='Ricoverati con sintomi')
plt.plot(df["terapia_intensiva"], 'k:', label='Terapia intensiva')
legend = ax.legend(loc='upper left', shadow=True)
plt.xlim(25,35)
```

```
plt.show()
```



Esercizio: cercare di riottenere nel modo più fedele possibile la figura seguente

<https://jakevdp.github.io/PythonDataScienceHandbook/04.09-text-and-annotation.html>

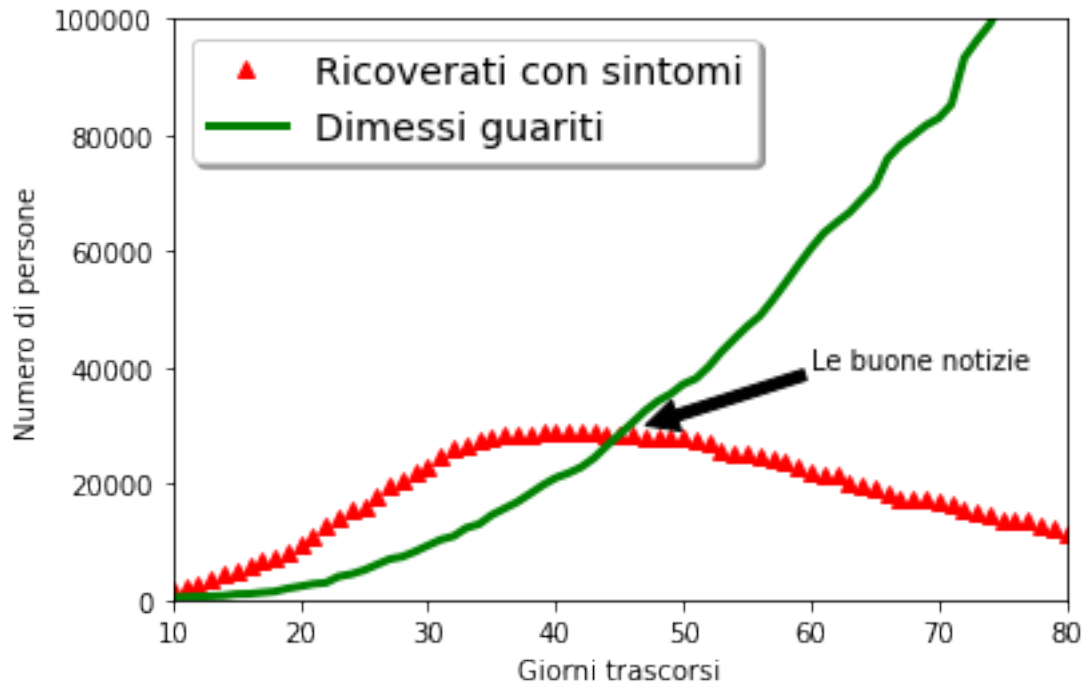
```
[ ]: from google.colab import files
from IPython.display import Image

uploaded = files.upload()
Image("Figura_esercizio.png", width=600)
```

<IPython.core.display.HTML object>

Saving Figura_esercizio.png to Figura_esercizio (5).png

```
[ ]:
```



```
[ ]: # Correzione esercizio
```

Esercizio: scegliere un paio di dati che si ritengono significativi e rappresentarli nella stessa immagine, “abbellendo” la figura in modo significativo (Divertitevi un po’!).

Esercizio: Scrivere un codice che permette di inserire uno accanto all’altro due grafici differenti (Ad. esempio nel primo i totale degli ospedalizzati e nel secondo accanto a destra coloro che sono in terapia intensiva)

Esercizio (Da inserire nella cartella *Esercizi terza lezione*): trovare un file.CSV a cui si è interessati in qualche sito di *open data* e fare una serie di grafici che rappresentino alcuni aspetti significativi.

3 Grafici con inserita come asse x la data

In questo caso si fa un plot con inserito l’asse x. Per mostrare le date “non tutte attaccate” si mostrano solo alcuni valori indicati con giorno e mese

```
[ ]: import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

df['data'] = pd.to_datetime(df['data'])
```

```

fig, ax = plt.subplots(figsize=(14,8))
plt.plot(df['data'],df['totale_casi'], 'o-') #,y='total_cases', data=df,
→label='Total cases', ax=ax, lw=2)
plt.ylabel('Units', fontsize=14)
plt.xlabel('Date', fontsize=14)
ax.xaxis.set_major_locator(mdates.AutoDateLocator())
ax.xaxis.set_major_formatter(mdates.DateFormatter('%d %m'))
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.legend(fontsize=12)
plt.title('Andamento casi totali (Italia)', fontsize=16)

```

[]: