



Progetto Covid-19

Autore : Renryan Maligaya
4^B informatica - Python

Download dei file

```
#funzione download file csv
def Download(url: str, dest_folder: str):
    #se la directory non esiste
    if not os.path.exists(dest_folder):
        #creo la directory
        os.makedirs(dest_folder)
    #rinomino il file
    nome_file = url.split("/")[-1].replace(".", "")
    #aggiungo il file all'interno della directory
    file_path = os.path.join(dest_folder, nome_file)
    #richiesta per scaricare
    richiesta = requests.get(url, stream=True)
    #se è andato a buon fine
    if richiesta.ok:
        #salvato in + percorso directory
        print("Download Success!")
        print("Saved in", os.path.abspath(file_path))
        with open(file_path, 'wb') as file:
            for chunk in richiesta.iter_content(chunk_size=1024*8):
                if chunk:
                    #scrittura nel file
                    file.write(chunk)
                    #svuota il buffer
                    file.flush()
                    #file.fileno restituisce un descrittore di file intero
                    os.fsync(file.fileno())
    else:
        #se il download non è andato a buon fine
        print("Download failed")
```

FUNZIONE : Download

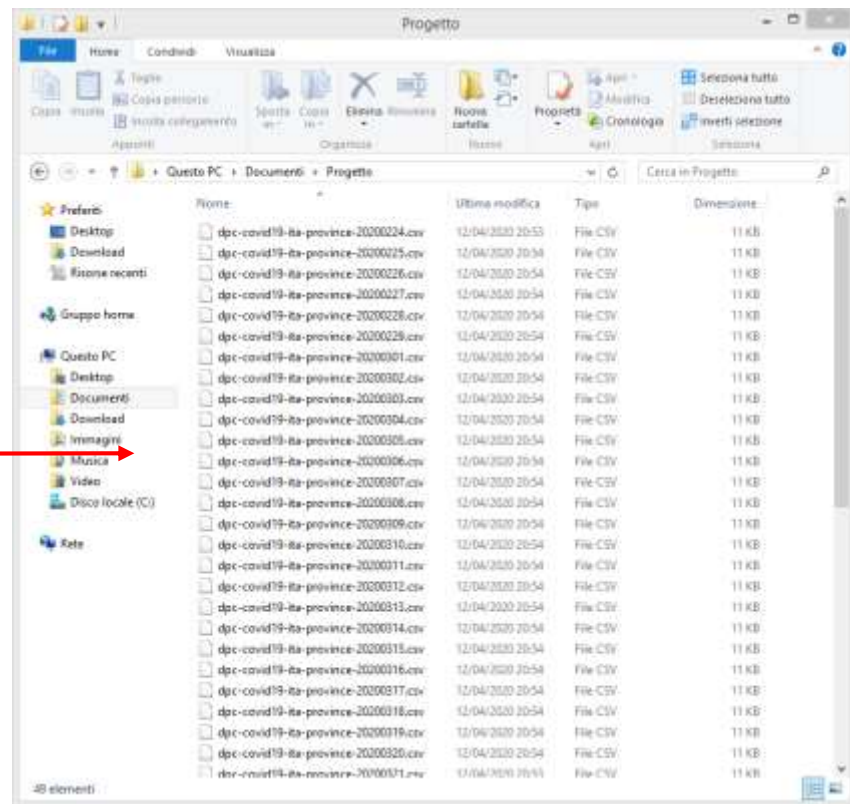
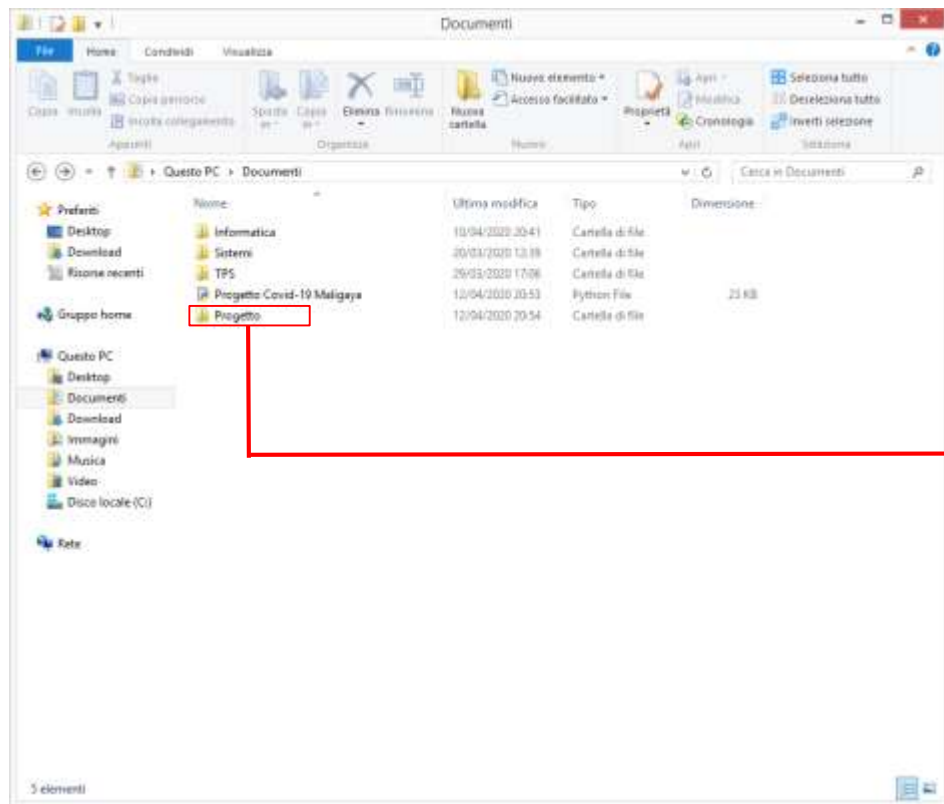
INPUT : address, cartella

OUTPUT : file csv scaricato

LIBRERIE : requests, os, path

Scarica il file csv dall'indirizzo passato del giorno attuale o quello indicato e lo inserisce nella directory(cartella) assegnata. Se la cartella non esiste, viene creata.

Cartella e file

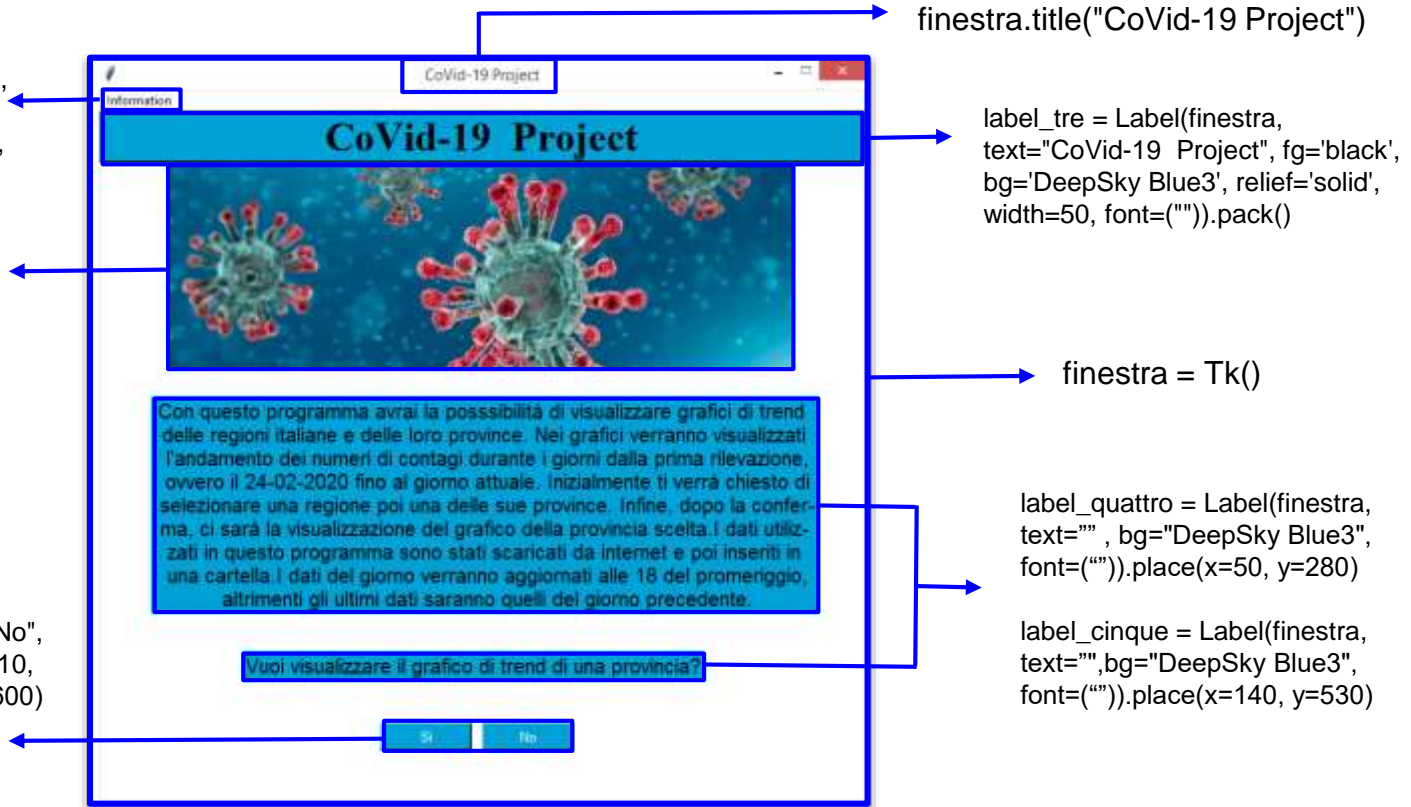


Pagina Iniziale - Home

```
submenu = Menu(menu)
menu.add_cascade(label="Information",
menu=submenu)
submenu.add_command(label="About",
command="")
```

```
img = Image.open()
my_img = ImageTk.PhotoImage(img)
lab_img = Label(image=my_img)
lab_img.pack()
```

```
button_uno = Button(finestra, text="Si/No",
fg='white', bg='DeepSky Blue3', width=10,
font=(), command="").place(x=275, y=600)
```



Pagina Informazioni



Selezione della Regione

```
finestra_reg = Tk()
```

```
immagine_it = Image.open("")  
ita_immagine =  
ImageTk.PhotoImage(immagine_it)  
it_immagine =  
Label(image=ita_immagine)  
it_immagine.place(x=0, y=40)
```

```
button_regione =  
Button(finestra_reg, text="Submit",  
fg='white', bg='orange', width=10,  
font=("",), command="").place(x=  
630, y= 730)
```



```
finestra_reg.title("CoVid-19 Project")
```

```
label_sel = Label(finestra_reg,  
text="", fg='black', bg='orange',  
relief='solid', width=60,  
font=("",)).pack()
```

```
cb1=Radiobutton(finestra_reg,  
text="Valle d'Aosta", fg="black",  
bg="green", variable="", font=("",),  
value=1).place(x=630, y=90)
```

```
cb8=Radiobutton(finestra_reg,  
text="Emilia-Romagna", fg="black",  
bg="white", variable="", font=("",),  
value =8).place(x=630, y=300)
```

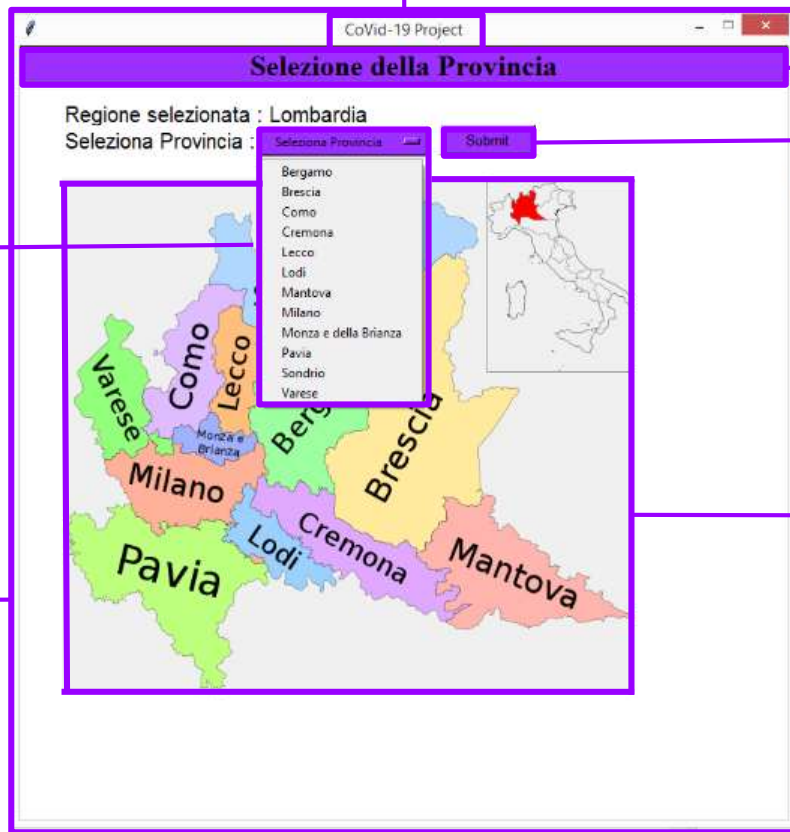
```
cb15=Radiobutton(finestra_reg,  
text="Abruzzo", fg="black",  
bg="red",variable="", font=("",),  
value=15).place(x= 630, y= 510)
```

Selezione della Provincia

```
finestra_prov.title("CoVid-19 Project")
```

```
droplist = OptionMenu(finestra_prov,  
var_provincia,*provincia)  
var_provincia.set("")  
droplist.config(width=20)  
droplist.config(bg = "purple1")  
droplist.place(x=225, y=75)
```

```
finestra_prov = Tk()
```

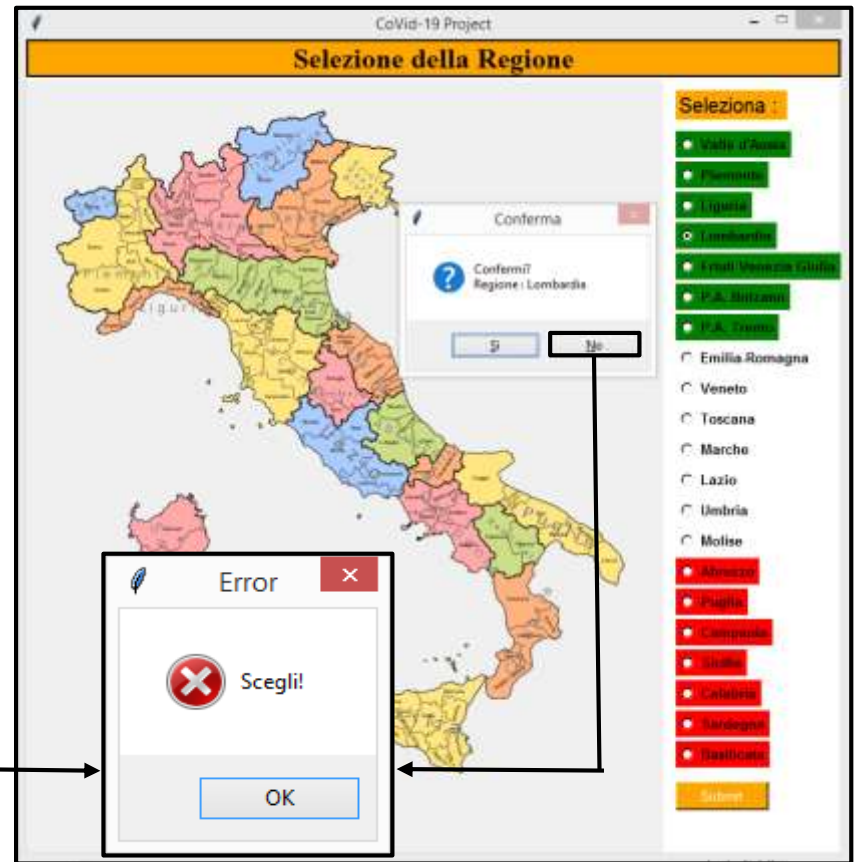
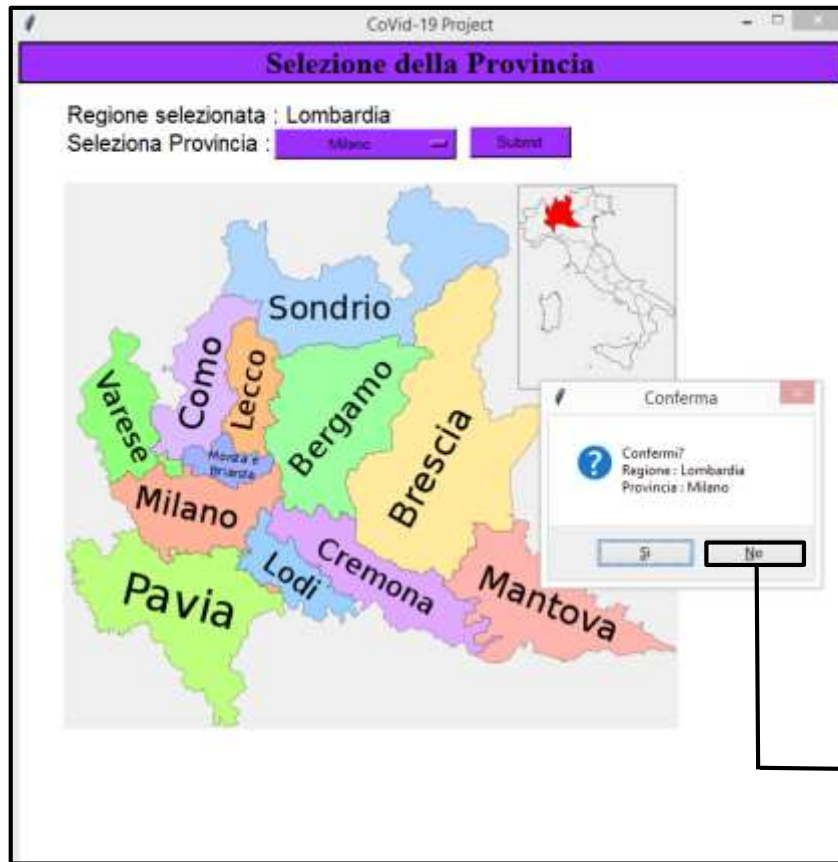


```
label_provincia =  
Label(finestra_prov,  
text="",fg='black', bg='purple1',  
relief='solid', width=60,  
font=(""))).pack()
```

```
button_provincia =  
Button(finestra_prov, text="Submit",  
fg='black', bg='purple1', width=10,  
font=(""))).place(x=  
400, y= 75)
```

```
photo = Image.open()  
my_photo =  
ImageTk.PhotoImage(photo)  
lab_photo =  
Label(image=my_photo)  
lab_photo.place(x=40, y= 125)
```

Conferma Regione e Provincia



Gestione e lettura dati file

```
#funzione di gestione e lettura dati file csv
def Daticsv(path: str):
    global provincia_finale

    #lista di tutti i dati
    all_dati = []
    #creo la lista con tutti i file
    lista_file = os.listdir(path)
    #ciclo for per la lunghezza della lista
    for i in range(len(lista_file)):
        #unisco la directory e il nome del file
        file_aprire = os.path.join(path, lista_file[i])
        #apro il file
        with open(file_aprire) as csvfile:
            #leggo il file csv
            reader = csv.DictReader(csvfile)
            reader = csvfile.readlines()
            struttura = reader[0].rstrip()
            #lista struttura file
            struttura = reader[0].split(',')
            for j in range(1:):
                #dati da una provincia di una regione in un giorno
                dati_riga = j.split(',')
                #inserisco nella lista i dati
                all_dati.append(dati_riga)

    #lista numero contagi
    contagi = []
    #lista date
    date = []
    if provincia_finale == "Friuli-Venezia":
        provincia_finale = "Friuli-Venezia"
    for k in range(len(all_dati)):
        #ricerca della regione e della provincia nel file
        if all_dati[k][1] == regione_finale and all_dati[k][5] == provincia_finale:
            c = int(all_dati[k][9])
            #inserisco nella lista il numero di contagi
            contagi.append(c)
            #creo il giorno
            g = [all_dati[k][0][1:10]].split('-')
            giorno = g[2] + "/" + g[1]
            #inserisco nella lista il giorno
            date.append(giorno)
    if provincia_finale == "Friuli-Venezia":
        provincia_finale = "Friuli-Venezia"

    Visualizza_Grafico(contagi, date)
```

FUNZIONE : Daticsv

INPUT : path(nome cartella)

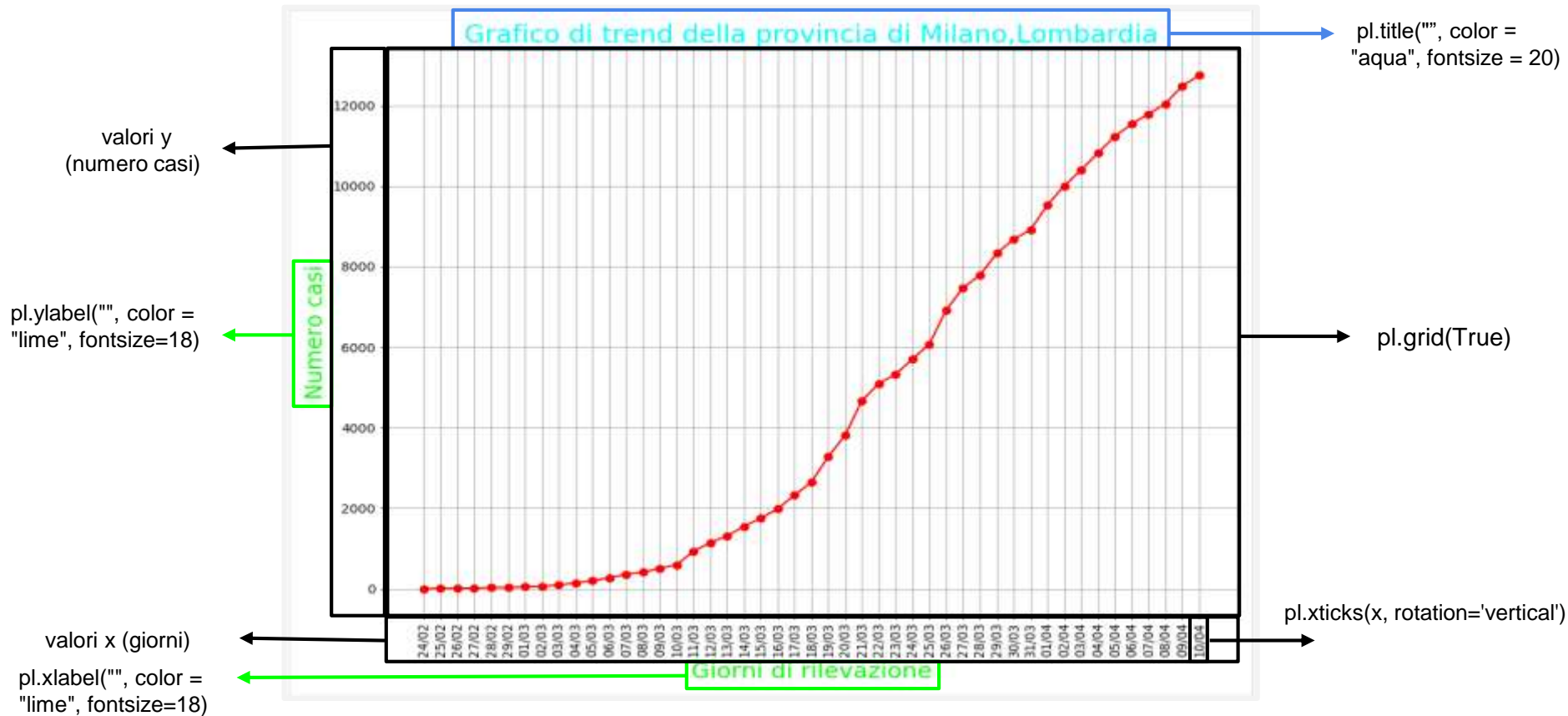
OUTPUT : all_dati, date, contagi

LIBRERIE : csv, os

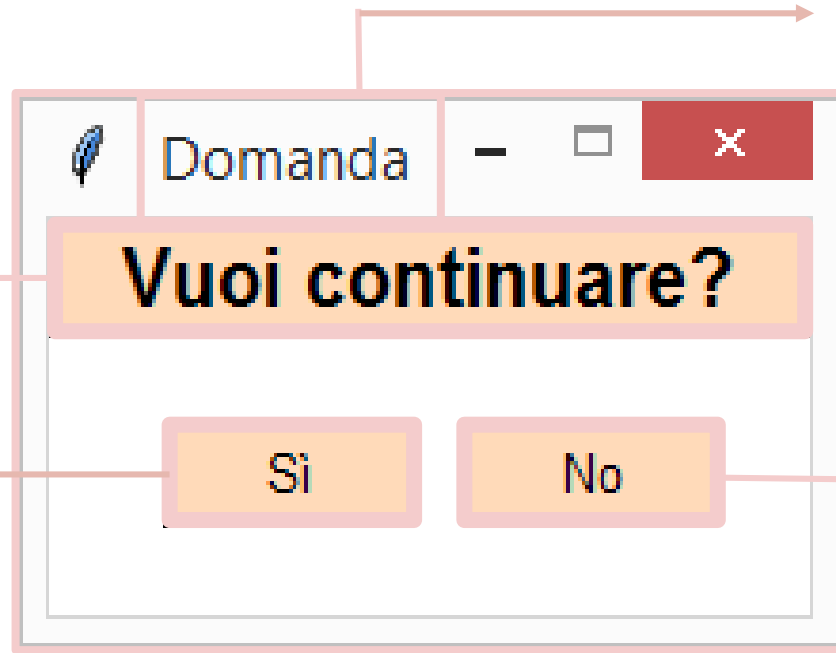
STRUTTURA FILE : Data, Regione, Provincia, numero di contagi

Grazie alle librerie è possibile leggere il file csv per ricavare i dati della provincia selezionata dall'utente per poi costruire successivamente il grafico di trend.

Visualizzazione del grafico



Richiesta utente



`finestra2.title("Domanda")`

`finestra2 = Tk()`

`label_risposta = Label(finestra2,
text="", fg='black', bg='peach puff',
relief='solid', width=20, font=("")) .pack()`

`button_si = Button(finestra2,
text="Si", fg='black', bg='peach
puff', width=7, font=("")) .place(x=30, y=50)`

`button_no = Button(finestra2,
text="No", fg='black', bg='peach
puff', width=7, font=("")) .place(x=110, y=50)`