Prova d'esame - Introduzione all'analisi dati con Python

June 15, 2023

---

1 Introduzione all'analisi dati con Python

2 Prova d'esame

Data: 12 giugno 2023

Autore: Michele Tizzoni

---

3 Istruzioni

La prova consiste in 5 esercizi elencati qui di seguito.

Ad ogni esercizio è assegnato un punteggio di 2 (due) punti. Due punti vengono assegnati per una soluzione completa e corretta. Un punto può essere assegnato se la soluzione è impostata correttamente o se l'esercizio è risolto parzialmente.

E' consigliato aggiungere commenti alle soluzioni proposte, in modo da spiegare che cosa dovrebbe fare il codice che scrivete.

La prova si considera superata con un punteggio finale di 6/10.

A prova conclusa, salvate il file con le vostre soluzione con il nome "Prova d'esame [Nome] [Cognome].ipynb".

4 Esercizio 1 - Lettura di file

Insieme al notebook trovate il file `cty_covariates.csv`. Questo file contiene i dati che vogliamo analizzare durante la prova. Si tratta di dati demografici e socio-economici degli Stati Uniti, riportati per ogni contea americana. Il dataset proviene dal sito Opportunity Atlas.

Richieste: 1. Usando la libreria `pandas` importare il file dentro un nuovo dataframe. 2. Mostrare a schermo le prime 5 righe del dataframe. 3. Stampare l'elenco delle variabili che rappresentano le colonne del dataframe.

[1]:
```
#scrivi qui la soluzione
import pandas as pd
```

```
df = pd.read_csv('cty_covariates.csv')
```

[3]:
```
#mostro a schermo le prime 5 righe del dataset
```

[2]:
```
df.head()
```

[2]:
```
   state  county       cz      czname  hhinc_mean2000  mean_commutetime2000
0      1       1  11101.0  Montgomery       74699.969             28.490602  \
1      1       3  11001.0      Mobile       76064.086             26.501080
2      1       5  10301.0     Eufaula       51246.004             24.047514
3      1       7  10801.0  Tuscaloosa       55094.492             32.875317
4      1       9  10700.0  Birmingham       62749.727             36.189240

   frac_coll_plus2000  frac_coll_plus2010  foreign_share2010  med_hhinc1990
0            0.189735            0.221990           0.020155   29718.635194  \
1            0.230036            0.260710           0.037592   26435.690624
2            0.107450            0.133496           0.028144   19026.749741
3            0.070026            0.099241           0.006859   19696.785014
4            0.096214            0.126334           0.047343   23159.691502

   …  singleparent_share1990  singleparent_share2000  traveltime15_2010
0  …                0.165540                0.240811           0.204163  \
1  …                0.184214                0.237883           0.275326
2  …                0.271470                0.393263           0.376049
3  …                0.188628                0.257294           0.252683
4  …                0.122455                0.173408           0.194344

    emp2000  mail_return_rate2010  ln_wage_growth_hs_grad  popdensity2010
0  0.609586             82.333183               -0.063314       91.802681  \
1  0.577026             80.034088                0.030093      114.647510
2  0.453271             74.899071                0.189366       31.029207
3  0.494241             70.003571               -0.020073       36.806339
4  0.577810             83.100349                0.096463       88.902191

   popdensity2000  ann_avg_job_growth_2004_2013  job_density_2013
0       73.466034                      0.010145         40.719135
1       88.323204                      0.012950         50.085987
2       32.818073                     -0.020756          9.230672
3       33.450962                     -0.004645         12.875392
4       79.148064                     -0.008120         36.175354

[5 rows x 35 columns]
```

[4]:
```
#stampo le colonne del dataframe
```

[5]:
```
df.columns
```

[5]:
```
Index(['state', 'county', 'cz', 'czname', 'hhinc_mean2000',
       'mean_commutetime2000', 'frac_coll_plus2000', 'frac_coll_plus2010',
       'foreign_share2010', 'med_hhinc1990', 'med_hhinc2016', 'poor_share2010',
       'poor_share2000', 'poor_share1990', 'share_white2010',
       'share_black2010', 'share_hisp2010', 'share_asian2010',
       'share_black2000', 'share_white2000', 'share_hisp2000',
       'share_asian2000', 'gsmn_math_g3_2013', 'rent_twobed2015',
       'singleparent_share2010', 'singleparent_share1990',
       'singleparent_share2000', 'traveltime15_2010', 'emp2000',
       'mail_return_rate2010', 'ln_wage_growth_hs_grad', 'popdensity2010',
       'popdensity2000', 'ann_avg_job_growth_2004_2013', 'job_density_2013'],
      dtype='object')
```

5 Esercizio 2 - Selezione e analisi

Dal dataframe precedente, selezionare la variable `med_hhinc2016` che rappresenta il reddito mediano delle famiglie residenti nell'anno 2016 e identificare (stampando a schermo):

1. La contea degli USA con il valore massimo di `med_hhinc2016`.
2. Il valore medio di `med_hhinc2016` in tutte le contee degli USA.

[6]:
```
#salvo in un dataframe la riga con il valore massimo di med_hhinc2016
df_max = df[df['med_hhinc2016']==df['med_hhinc2016'].max()]
```

[22]:
```
#stampo il nome della contea con il massimo reddito mediano familiare
df_max['czname']
```

[22]:
```
2872    Washington DC
Name: czname, dtype: object
```

[9]:
```
#stampo la media della variabile med_hhinc2016 su tutte le contee degli USA
df['med_hhinc2016'].mean()
```

[9]:
```
48259.86991227927
```

6 Esercizio 3 - Visualizzazione

Usando la libreria `matplotlib` creare una figura che mostri la relazione esistente tra due variabili del dataset importato. In particolare:

1. Disegnare un grafico che mostri il reddito mediano famigliare annuo nel 2016 (`med_hhinc2016`) sull'asse y e la percentuale di popolazione afro-americana nel 2010 (`share_black2010`) sull'asse x. I valori devono essere visualizzati come punti blu.
2. Aggiungere le seguenti etichette sugli assi: `household median income 2016 ($)` per l'asse y, `share of black population (2010)` per l'asse x.

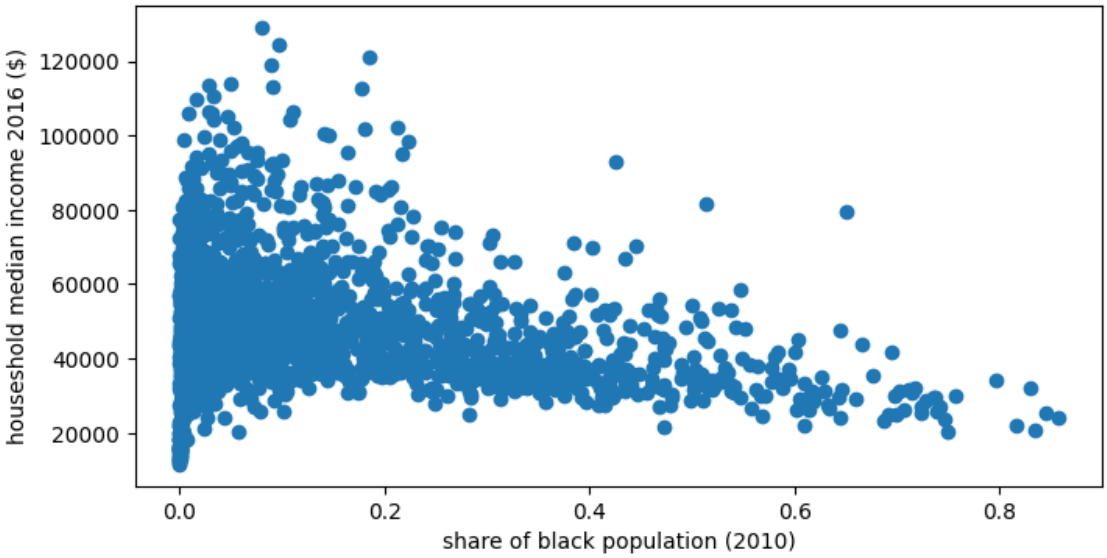
[10]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

[13]:
```
plt.figure(figsize=(8,4))

plt.plot(df.share_black2010, df.med_hhinc2016, 'o')

plt.xlabel('share of black population (2010)')
plt.ylabel('houseshold median income 2016 ($)')
```

[13]:
```
Text(0, 0.5, 'houseshold median income 2016 ($)')
```



7 Esercizio 4 - Controllo di flusso

Dato il seguente dizionario:

`states={'California': 6, 'Georgia': 13, 'Maine': 23}`

che assegna ad ogni stato americano un codice univoco, come quello presente nel dataset, usare un ciclo `for`, per ogni stato presente come chiave del dizionario e: 1. selezionare nel dataset solo le contee di quello stato (identificato dalla variable numerica `state`); 2. calcolare il valore medio di `med_hhinc2016` in tutte le contee di quello stato; 2. stampare il nome dello stato e il valore medio di `med_hhinc2016`.

[14]:
```
states={'California': 6, 'Georgia': 13, 'Maine':23}
```

[21]:
```
#inizio il ciclo for sulle chiavi del dizionario
for s in states:
```

```
    state_code = states[s] #il codice dello stato è il valore associato a ogni␣
↪chiave

    df_state = df[df['state']==state_code] #seleziono solo le contee dello␣
↪stato s

    avg_income = df_state['med_hhinc2016'].mean() #calcolo la media

    print('In',s,'the average household income in 2016 was')
    print(avg_income,'USD')
    print('--')
```

```
In California the average household income in 2016 was
60779.70015192415 USD
--
In Georgia the average household income in 2016 was
42518.241208151536 USD
--
In Maine the average household income in 2016 was
48296.17271210717 USD
--
```

8 Esercizio 5 - Funzioni

Scrivere una funzione chiamata `analyze_state` che prende come **input** 3 variabili:

1. il dataframe con il dataset che stiamo analizzando;
2. il dizionario states definito nell'esercizio 4;
3. una stringa che rappresenta il nome di uno stato presente come chiave nel dizionario;

e restituisce come **output** la seguente scritta: 1. "Nel 2010, la percentuale media di popolazione afro-americana in STATE era AVGBLACK"

dove `STATE` è il nome dello stato e `AVGBLACK` è la media della variabile `share_black2010` nelle contee di quello stato.

Verificare che la funzione dia il risultato atteso per lo stato 'California'.

[19]:
```
def analyze_state(df, dizionario, STATE):

    df_state = df[df['state']==dizionario[STATE]]

    AVGBLACK = df_state['share_black2010'].mean()

    print('Nel 2010, la percentuale media di popolazione afro-americana␣
↪in',STATE,'era',AVGBLACK)
```

[20]:
```
analyze_state(df, states, 'California')
```

```
Nel 2010, la percentuale media di popolazione afro-americana in California era
0.0362626999087931
```

[ ]:
```

```