

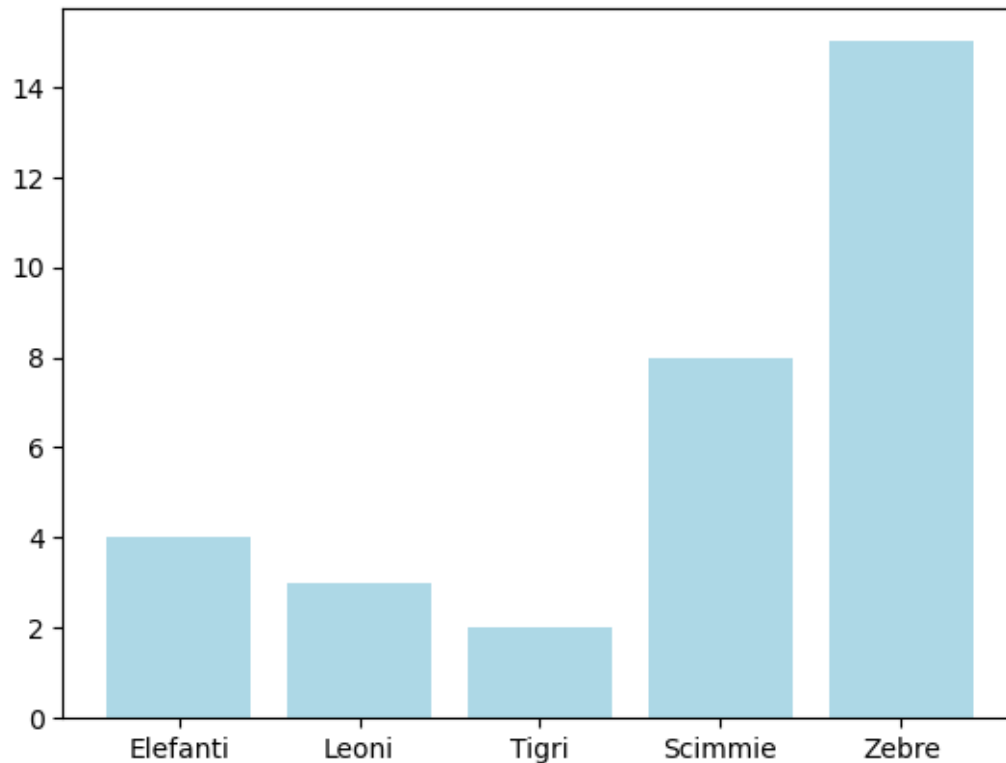
Grafici Robotica Esercitazione3

December 11, 2023

1 Grafici

1.1 Grafico a Barre della Distribuzione del Numero di Animali nello Zoo

```
[2]: # Codice per generare un diagramma a barre che visualizza la distribuzione del  
      ↪ numero di animali in uno zoo  
import matplotlib.pyplot as plt  
  
# Dati relativi al numero di animali  
animali = ['Elefanti', 'Leoni', 'Tigri', 'Scimmie', 'Zebre']  
numero_animali = [4, 3, 2, 8, 15]  
  
# Creazione del diagramma a barre con colorazione chiara  
plt.bar(animali, numero_animali, color="lightblue")  
  
# Visualizzazione del diagramma  
plt.show()
```



1.2 Grafico a Barre della Distribuzione del Numero di Animali nello Zoo

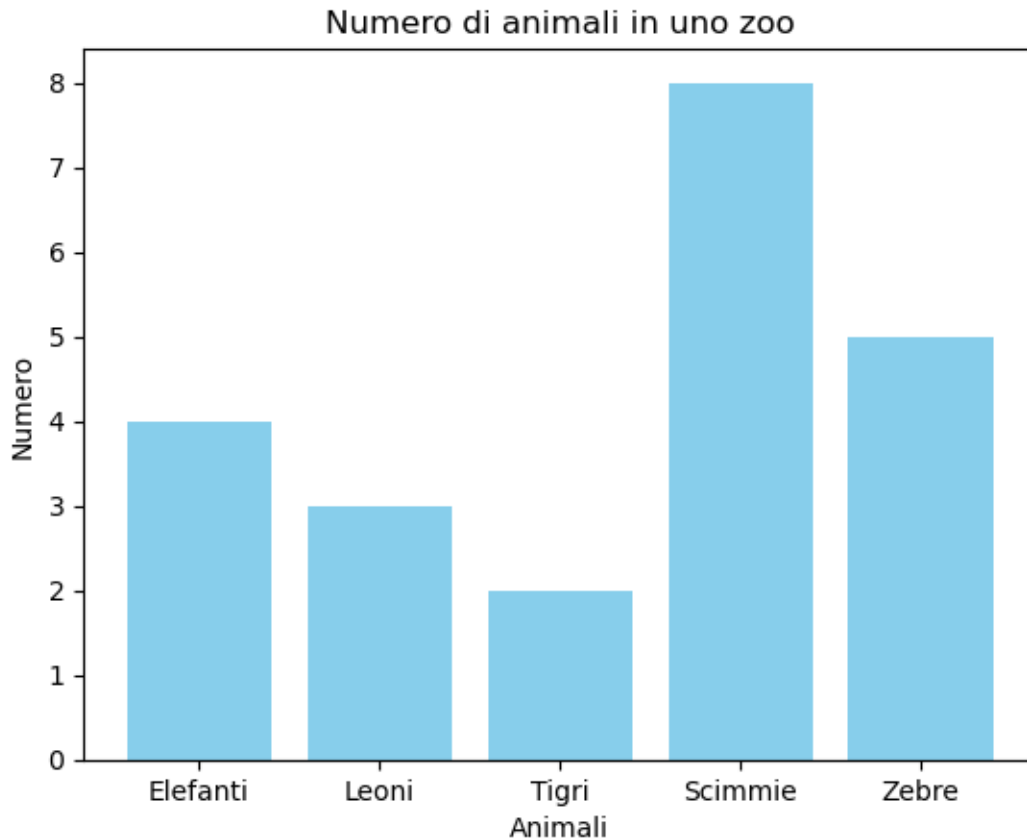
```
[4]: # Codice per generare un grafico a barre che visualizza la distribuzione del
      ↪ numero di animali in uno zoo
import matplotlib.pyplot as plt

# Dati relativi al numero di animali
animali = ['Elefanti', 'Leoni', 'Tigri', 'Scimmie', 'Zebre']
numero_animali = [4, 3, 2, 8, 5]

# Creazione del grafico a barre con colorazione chiara
plt.bar(animali, numero_animali, color='skyblue')

# Aggiunta di titolo e etichette agli assi
plt.title('Numero di animali in uno zoo')
plt.xlabel('Animali')
plt.ylabel('Numero')

# Visualizzazione del grafico
plt.show()
```



1.3 Grafico a Linee - Andamento delle Temperature Medie Mensili nel Corso dell'Anno

```
[5]: # Codice per generare un grafico a linee che visualizza l'andamento delle
      ↳ temperature medie mensili
import matplotlib.pyplot as plt

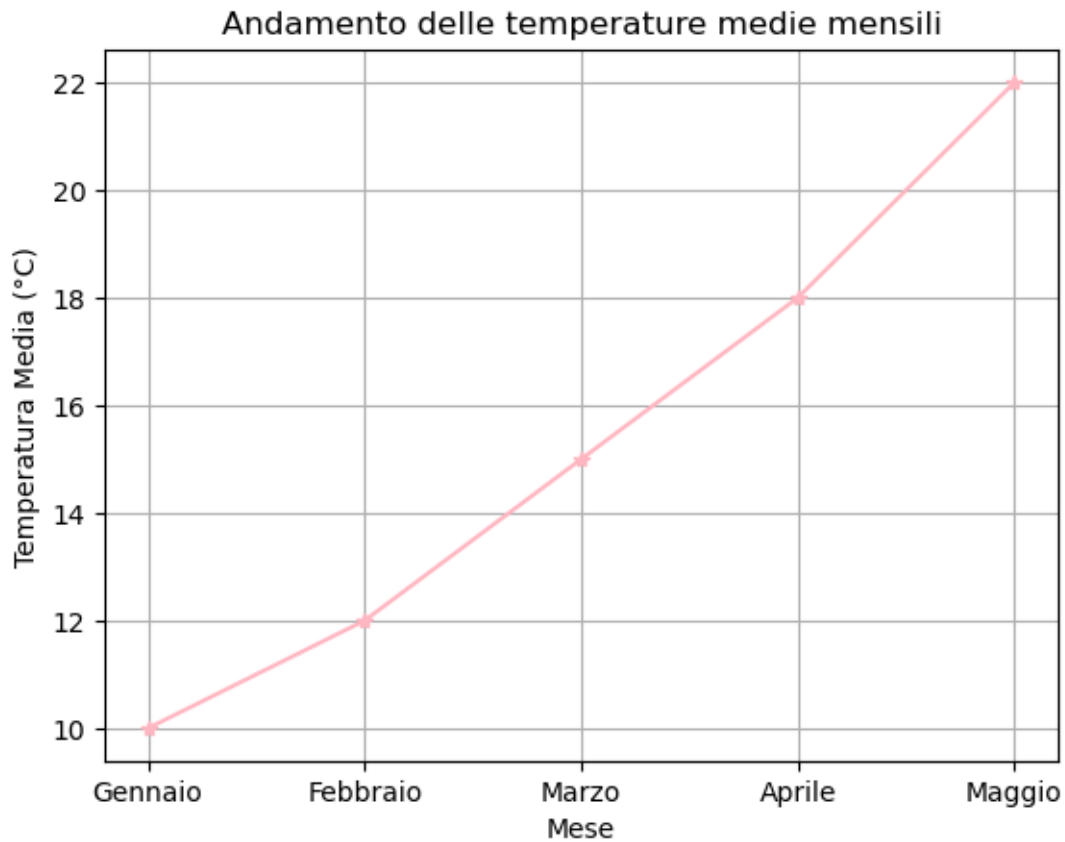
# Dati relativi alle temperature medie mensili
mese = ['Gennaio', 'Febbraio', 'Marzo', 'Aprile', 'Maggio']
temperatura_media = [10, 12, 15, 18, 22]

# Creazione del grafico a linee con marcatori a forma di stella e linea di
↳ connessione
plt.plot(mese, temperatura_media, marker='*', linestyle='-', color='lightpink')

# Aggiunta di titolo e etichette agli assi
plt.title('Andamento delle temperature medie mensili')
plt.xlabel('Mese')
plt.ylabel('Temperatura Media (°C)')
```

```
# Attivazione delle linee guida sulla griglia per una migliore lettura
plt.grid(True)

# Visualizzazione del grafico
plt.show()
```



1.4 Grafico a Linee Tratteggiate - Andamento delle Temperature Medie Mensili nel Corso dell'Anno

```
[8]: # Codice per generare un grafico a linee che visualizza l'andamento delle
      ↳ temperature medie mensili
import matplotlib.pyplot as plt

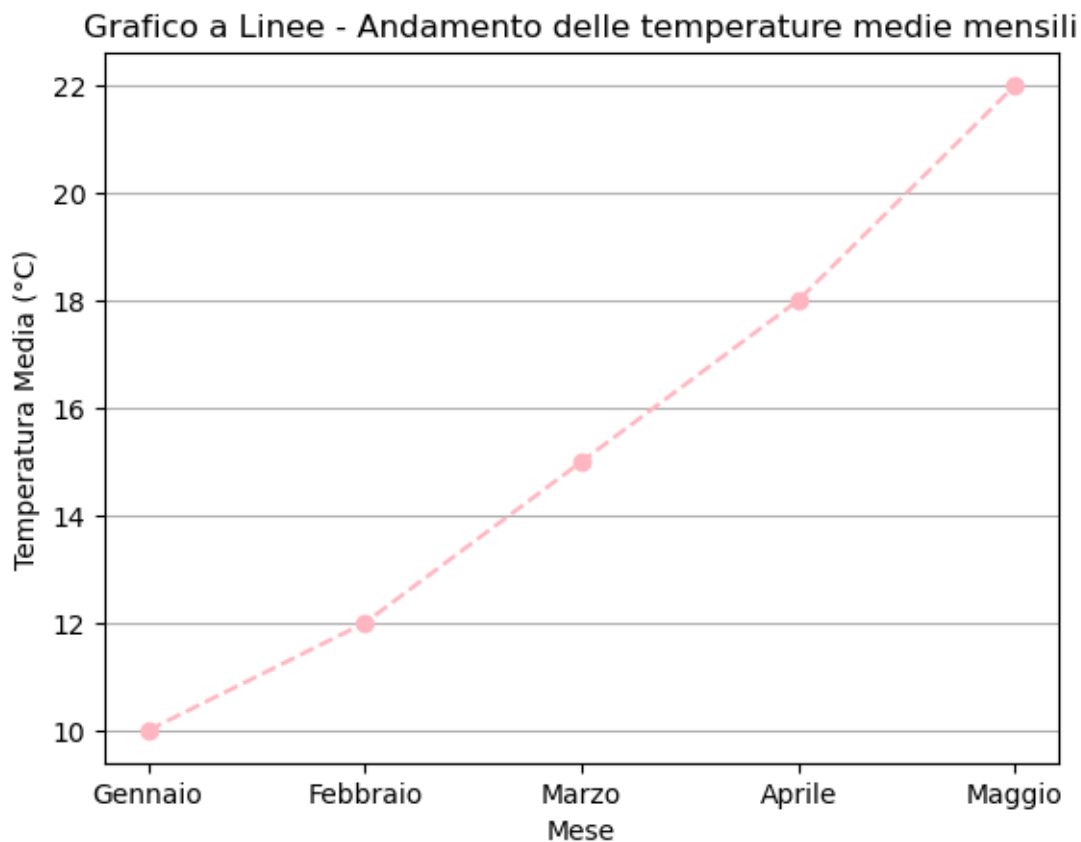
# Dati relativi alle temperature medie mensili
mese = ['Gennaio', 'Febbraio', 'Marzo', 'Aprile', 'Maggio']
temperatura_media = [10, 12, 15, 18, 22]

# Creazione del grafico a linee con marcatori circolari e stile tratteggiato
plt.plot(mese, temperatura_media, marker='o', linestyle='--', color='lightpink')
```

```
# Aggiunta di titolo e etichette agli assi
plt.title('Grafico a Linee - Andamento delle temperature medie mensili')
plt.xlabel('Mese')
plt.ylabel('Temperatura Media (°C)')

# Attivazione delle linee guida solo sull'asse y per una migliore lettura
plt.grid(True, axis="y")

# Visualizzazione del grafico
plt.show()
```



1.5 Grafico a Linee Tratteggiate - Andamento delle Temperature Medie Mensili nel Corso dell'Anno

```
[9]: # Codice per generare un grafico a linee che visualizza l'andamento delle
      ↳ temperature medie mensili
import matplotlib.pyplot as plt

# Dati relativi alle temperature medie mensili
mese = ['Gennaio', 'Febbraio', 'Marzo', 'Aprile', 'Maggio']
```

```

temperatura_media = [10, 12, 15, 18, 22]

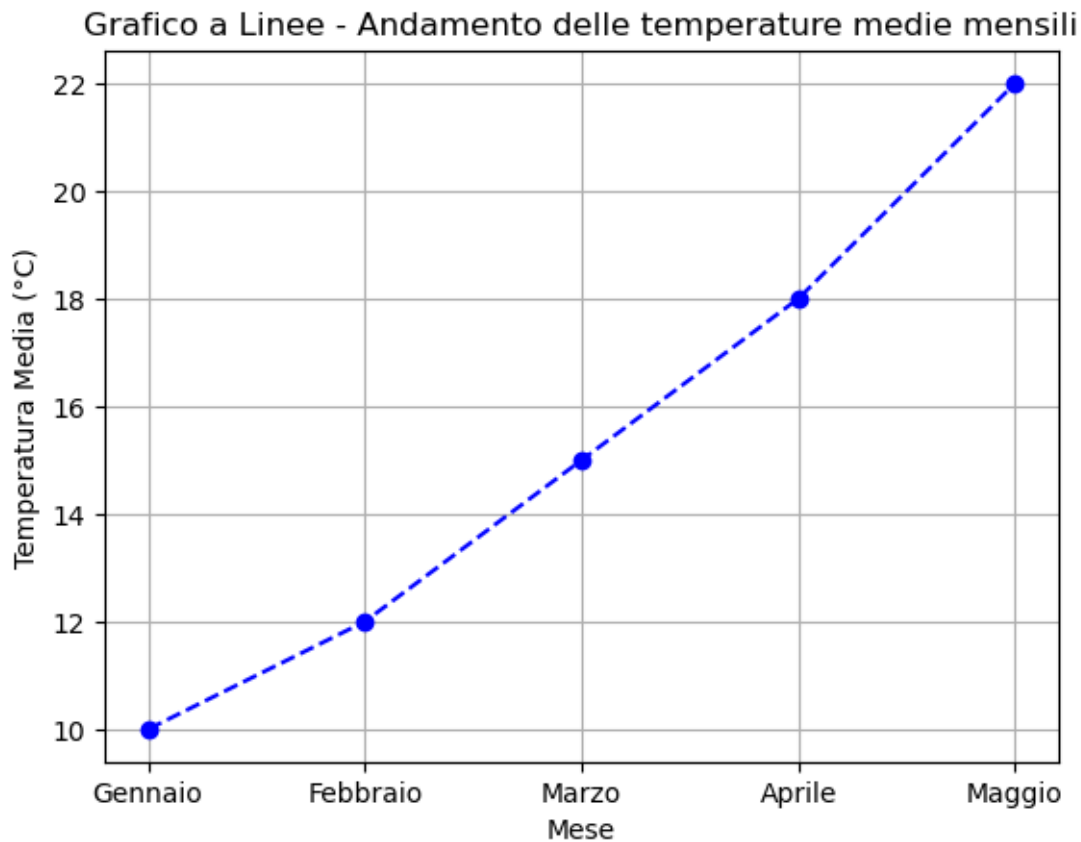
# Creazione del grafico a linee con marcatori circolari, stile tratteggiato e
→ colore blu
plt.plot(mese, temperatura_media, marker='o', linestyle='--', color='blue')

# Aggiunta di titolo e etichette agli assi
plt.title('Grafico a Linee - Andamento delle temperature medie mensili')
plt.xlabel('Mese')
plt.ylabel('Temperatura Media (°C)')

# Attivazione delle linee guida sulla griglia per una migliore lettura
plt.grid(True)

# Visualizzazione del grafico
plt.show()

```



1.6 Grafico a Barre - Vendite Mensili

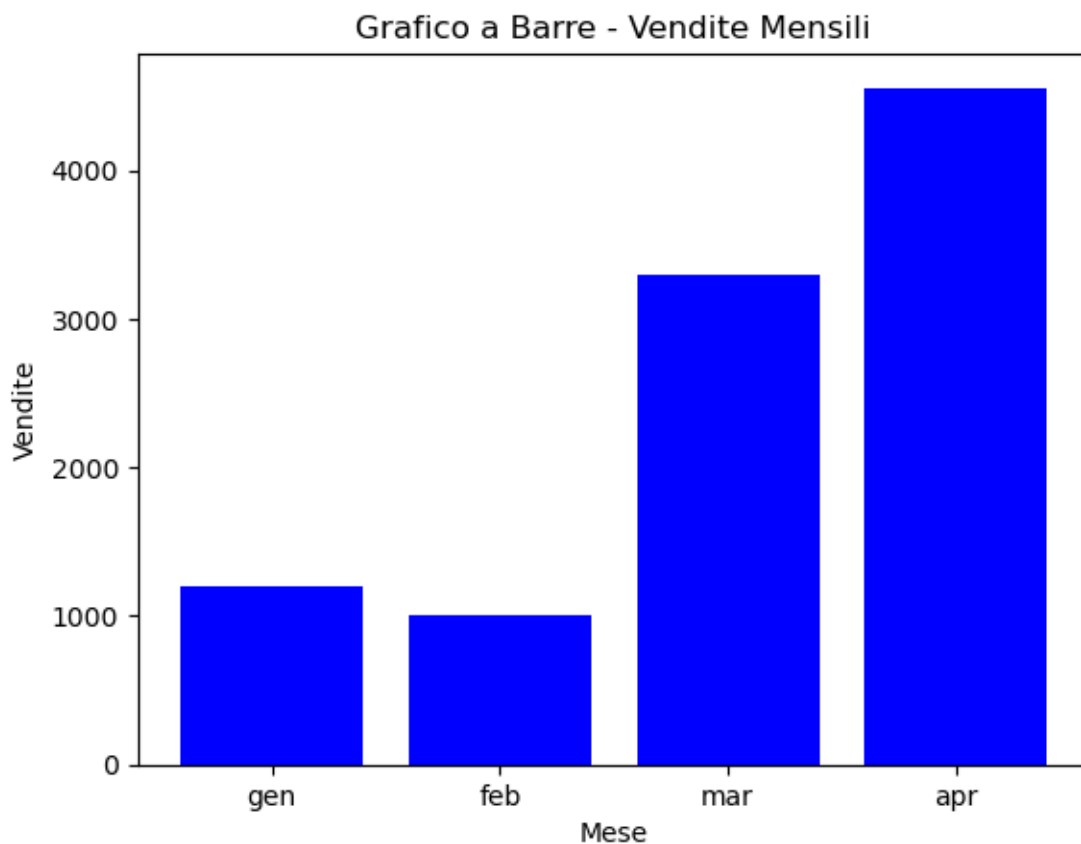
```
[10]: # Codice per generare un grafico a barre che visualizza le vendite mensili
import matplotlib.pyplot as plt

# Dati relativi alle vendite mensili
vendite_mensili = {
    "gen": 1200,
    "feb": 1000,
    "mar": 3300,
    "apr": 4555
}

# Creazione del grafico a barre con colore blu
plt.bar(vendite_mensili.keys(), vendite_mensili.values(), color="blue")

# Aggiunta di titolo e etichette agli assi
plt.title('Grafico a Barre - Vendite Mensili')
plt.xlabel('Mese')
plt.ylabel('Vendite')

# Visualizzazione del grafico
plt.show()
```



1.7 Grafico a Torta - Percentuale di Temperatura Media Mensile

```
[11]: # Codice per generare un grafico a torta che mostra la percentuale di
      ↳ temperatura media mensile
import matplotlib.pyplot as plt

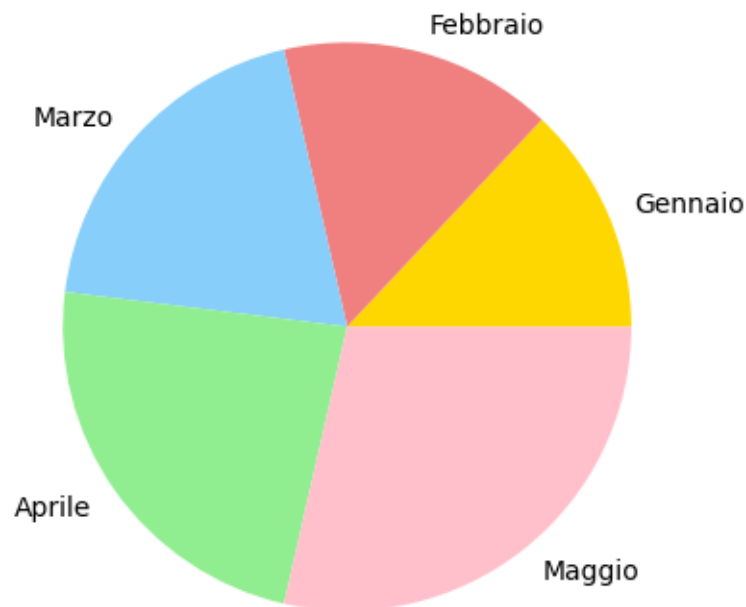
# Dati relativi alle temperature medie mensili, colori e mesi
temperatura_media = [10, 12, 15, 18, 22]
colori = ['gold', 'lightcoral', 'lightskyblue', 'lightgreen', 'pink']
mese = ['Gennaio', 'Febbraio', 'Marzo', 'Aprile', 'Maggio']

# Creazione del grafico a torta con etichette dei mesi e colori specifici
plt.pie(temperatura_media, labels=mese, colors=colori)

# Aggiunta di titolo al grafico
plt.title('Grafico a Torta - Percentuale di temperatura media mensile')

# Visualizzazione del grafico
plt.show()
```

Grafico a Torta - Percentuale di temperatura media mensile



1.8 Grafico a Torta - Percentuale di Temperatura Media Mensile

```
[12]: # Codice per generare un grafico a torta che mostra la percentuale di
      ↳ temperatura media mensile
import matplotlib.pyplot as plt

# Dati relativi alle temperature medie mensili, colori e nomi dei mesi
temperatura_mesi = {
    'Gennaio': 10,
    'Febbraio': 12,
    'Marzo': 15,
    'Aprile': 18,
    'Maggio': 22
}

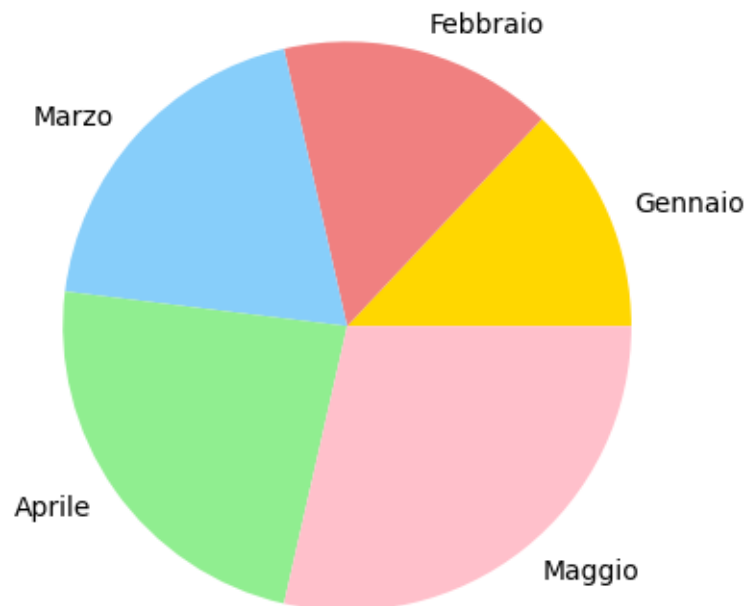
colori = ['gold', 'lightcoral', 'lightskyblue', 'lightgreen', 'pink']

# Creazione del grafico a torta con etichette dei mesi e colori specifici
plt.pie(temperatura_mesi.values(), labels=temperatura_mesi.keys(), colors=colori)

# Aggiunta di titolo al grafico
plt.title('Grafico a Torta - Percentuale di temperatura media mensile')

# Visualizzazione del grafico
plt.show()
```

Grafico a Torta - Percentuale di temperatura media mensile



1.9 Scatter Plot - Età vs Altezza

```
[13]: # Codice per generare uno scatter plot che mostra la relazione tra età e altezza
import matplotlib.pyplot as plt

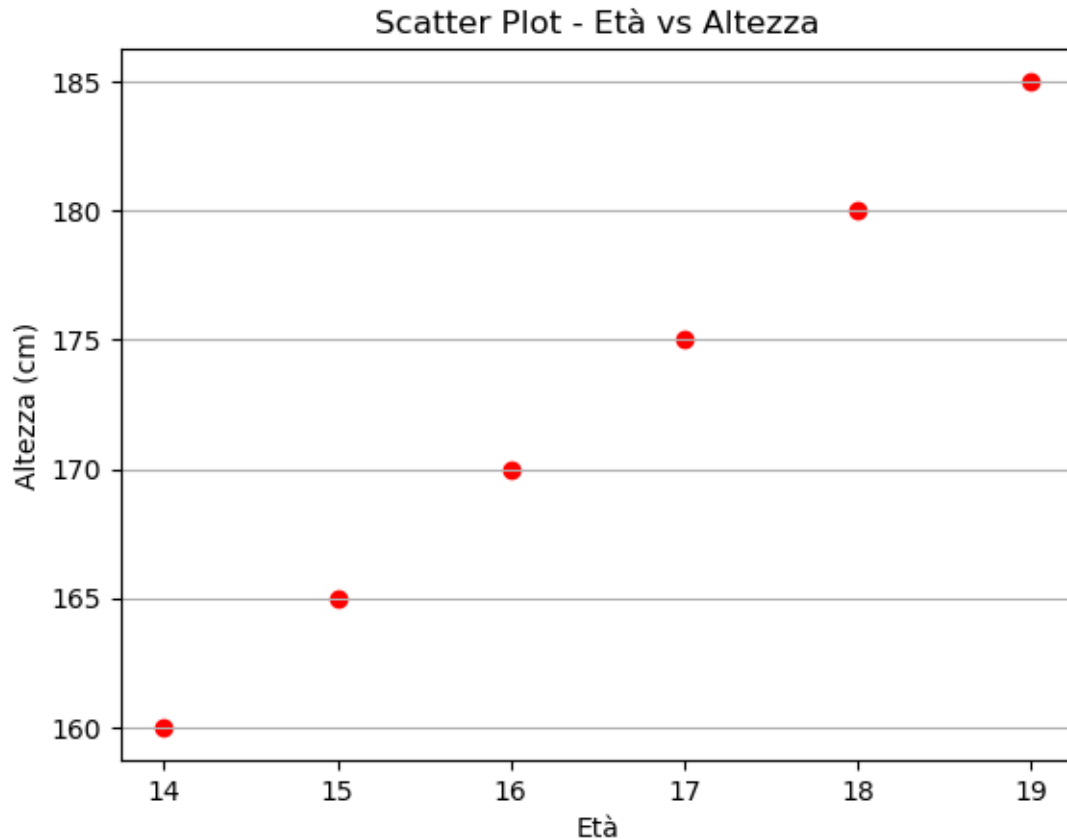
# Dati relativi a età e altezza
età = [14, 15, 16, 17, 18, 19]
altezza = [160, 165, 170, 175, 180, 185]

# Creazione dello scatter plot con cerchi rossi
plt.scatter(età, altezza, color='red', marker='o') # In questo caso marker, 'o'
↳ indica l'uso di cerchi come marcatori

# Aggiunta di titolo e etichette agli assi
plt.title('Scatter Plot - Età vs Altezza')
plt.xlabel('Età')
plt.ylabel('Altezza (cm)')

# Attivazione delle linee guida sulla griglia sull'asse y per una migliore
↳ lettura
plt.grid(True, axis='y')

# Visualizzazione dello scatter plot
plt.show()
```



1.10 Grafico a Barre Orizzontali - Punteggi degli Studenti

```
[15]: # Codice per generare un grafico a barre orizzontali che mostra i punteggi degli
      ↪ studenti
import matplotlib.pyplot as plt

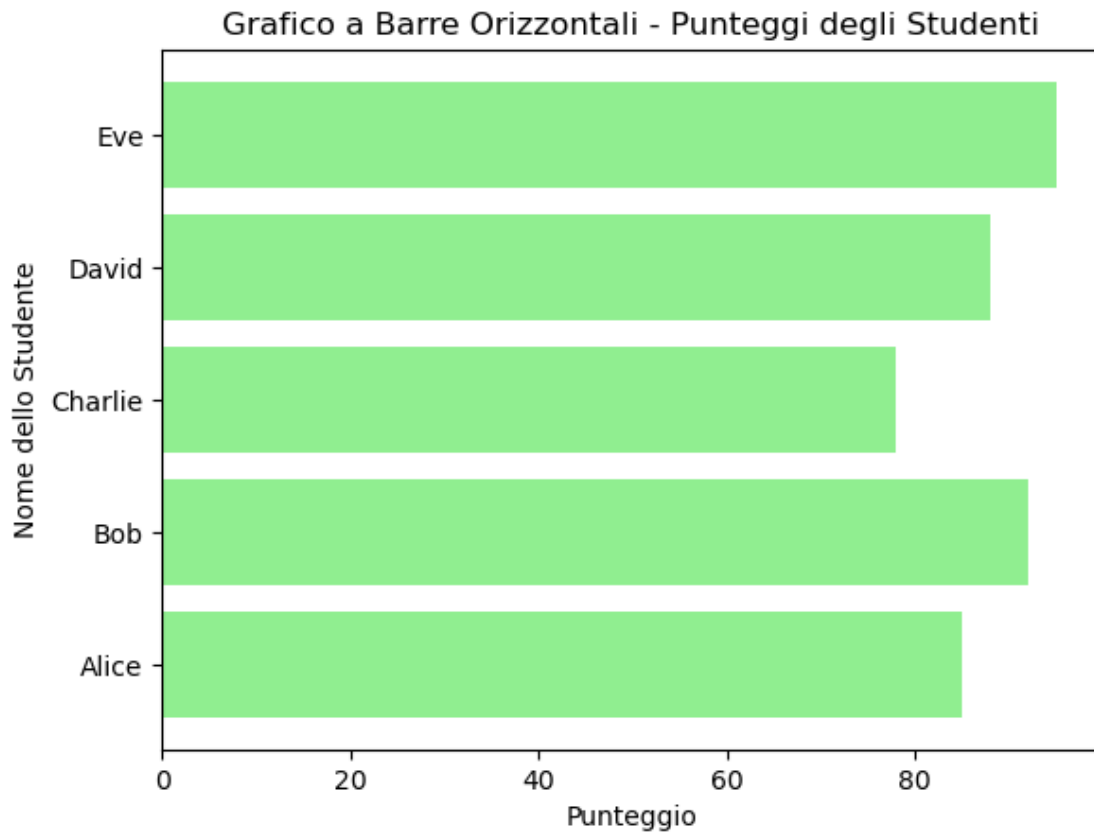
# Dati relativi ai nomi degli studenti e ai loro punteggi
nomi_studenti = ['Alice', 'Bob', 'Charlie', 'David', 'Eve']
punteggi = [85, 92, 78, 88, 95]

# Creazione del grafico a barre orizzontali con colori verdi chiari
plt.barh(nomi_studenti, punteggi, color='lightgreen')

# Aggiunta di titolo e etichette agli assi
plt.title('Grafico a Barre Orizzontali - Punteggi degli Studenti')
plt.xlabel('Punteggio')
plt.ylabel('Nome dello Studente')

# Visualizzazione del grafico
```

```
plt.show()
```



1.11 DataFrame - Punteggi degli Studenti (Ordinati per Punteggio)

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

# Dati relativi ai nomi degli studenti e ai loro punteggi
nomi_studenti = ['Alice', 'Bob', 'Charlie', 'David', 'Eve']
punteggi = [85, 92, 78, 88, 95]

# Crea un DataFrame utilizzando pandas
data = {'Nome dello Studente': nomi_studenti, 'Punteggio': punteggi}
df = pd.DataFrame(data)

# Ordina il DataFrame per punteggio in ordine crescente
df.sort_values(by='Punteggio', inplace=True)

# Visualizza il DataFrame ordinato
df
```

```
[18]:
```

	Nome dello Studente	Punteggio
2	Charlie	78
0	Alice	85
3	David	88
1	Bob	92
4	Eve	95

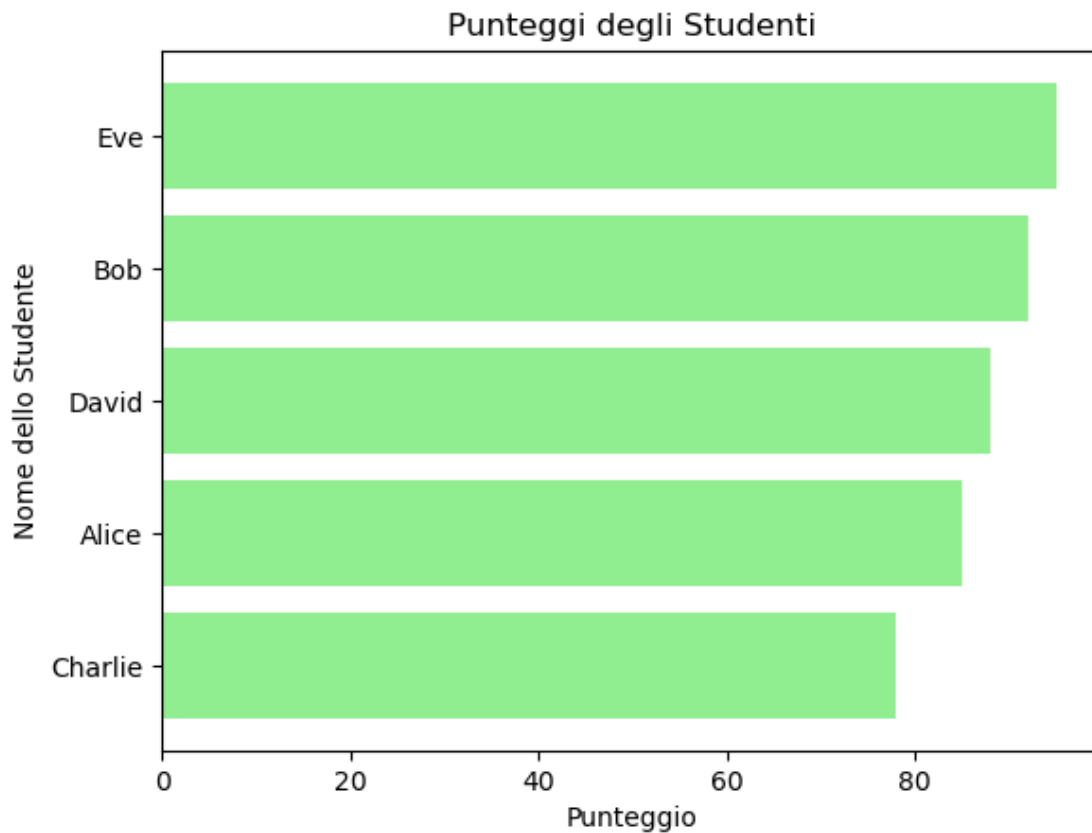
1.12 Grafico a Barre Orizzontali - Punteggi degli Studenti

```
[19]: # Crea un grafico a barre orizzontali utilizzando i dati dal DataFrame ordinato
plt.barh(df['Nome dello Studente'], df['Punteggio'], color='lightgreen')

# Aggiunge un titolo al grafico
plt.title('Punteggi degli Studenti')

# Etichette degli assi x e y
plt.xlabel('Punteggio')
plt.ylabel('Nome dello Studente')

# Visualizza il grafico
plt.show()
```



1.13 Grafico a Barre Orizzontali - Punteggi degli Studenti

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

nomi_studenti = ['Alice', 'Bob', 'Charlie', 'David', 'Eve']
punteggi = [85, 92, 78, 88, 95]

# Crea un DataFrame con nomi e punteggi
data = {'Nome dello Studente': nomi_studenti, 'Punteggio': punteggi}
df = pd.DataFrame(data)

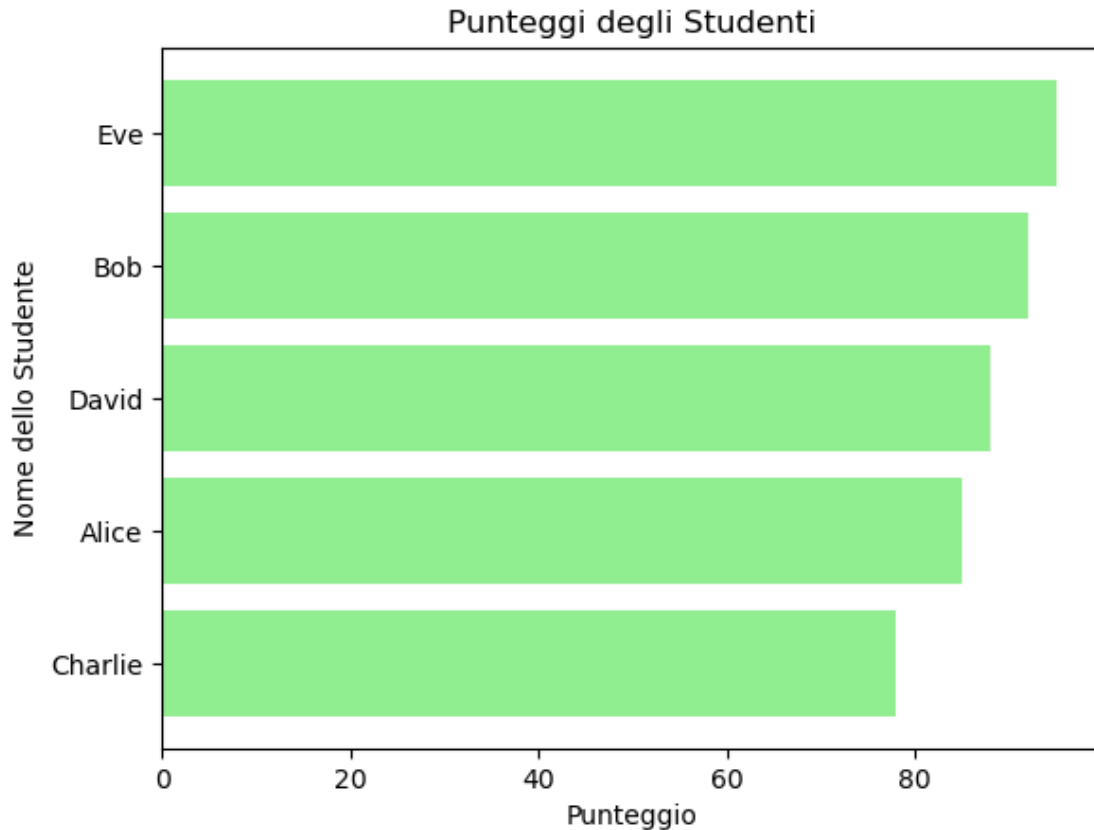
# Ordina il DataFrame per punteggio in ordine crescente e crea un nuovo
↳ DataFrame ordinato
df_sorted = df.sort_values(by='Punteggio')

# Crea un grafico a barre orizzontali utilizzando i dati dal DataFrame ordinato
plt.barh(df_sorted['Nome dello Studente'], df_sorted['Punteggio'],
↳ color='lightgreen')

# Aggiunge un titolo al grafico
plt.title('Punteggi degli Studenti')

# Etichette degli assi x e y
plt.xlabel('Punteggio')
plt.ylabel('Nome dello Studente')

# Visualizza il grafico
plt.show()
```



1.14 Grafico a Dispersione Altezza vs Peso

```
[19]: import matplotlib.pyplot as plt
import numpy as np

# Dati di esempio
altezza_maschi = np.random.normal(175, 10, 50) # NumPy l'altezza ha una media
↳ di 175 cm con una deviazione standard di 10 cm, e il peso ha una media di 70
↳ kg con una deviazione standard di 5 kg.
peso_maschi = np.random.normal(70, 5, 50) # Peso dei maschi

altezza_femmine = np.random.normal(162, 8, 50) # Altezza delle femmine
peso_femmine = np.random.normal(58, 4, 50) # Peso delle femmine

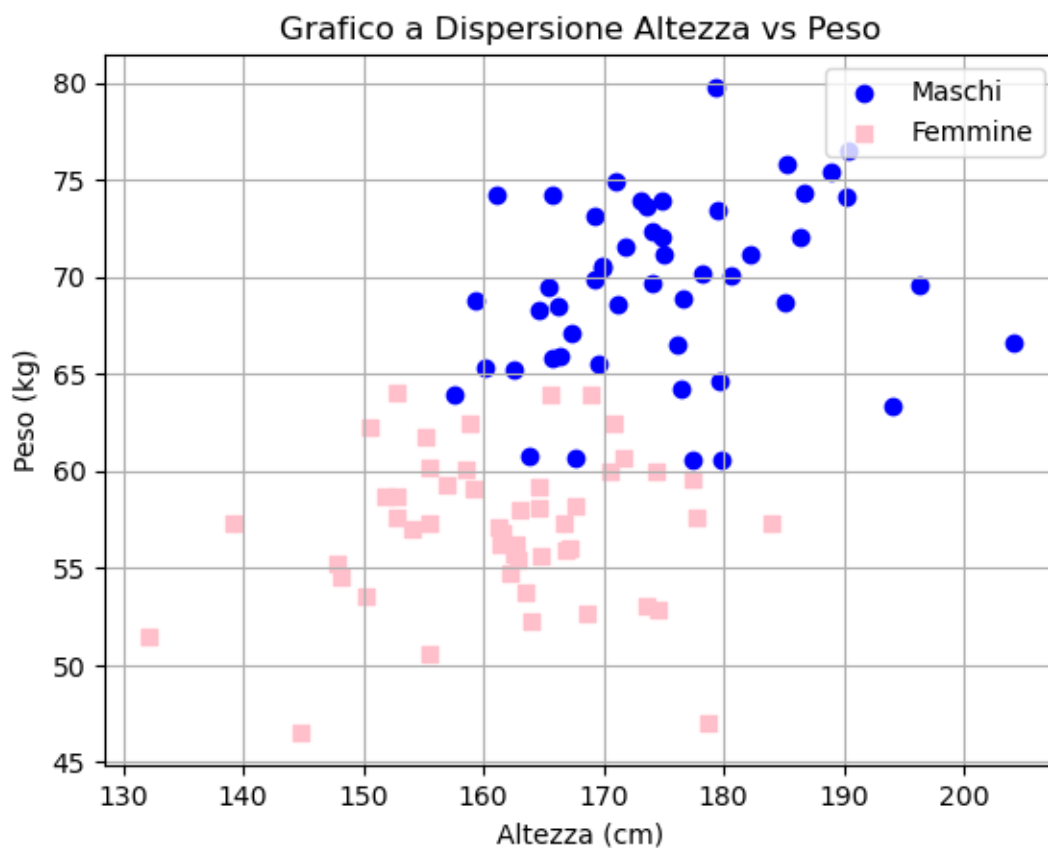
# Crea il grafico a dispersione per i maschi
plt.scatter(altezza_maschi, peso_maschi, color='blue', label='Maschi',
↳ marker='o')

# Crea il grafico a dispersione per le femmine
```

```
plt.scatter(altezza_femmine, peso_femmine, color='pink', label='Femmine',
            ↪marker='s')

# Personalizza il grafico
plt.title('Grafico a Dispersione Altezza vs Peso')
plt.xlabel('Altezza (cm)')
plt.ylabel('Peso (kg)')
plt.legend(loc='upper right') # Aggiunge la legenda nell'angolo in alto a destra
plt.grid(True) # Aggiunge una griglia al grafico

# Mostra il grafico
plt.show()
```



1.15 Grafico a Barre Apilato per Materia e Sesso

```
[18]: # Dati di esempio
materie = ['Matematica', 'Scienze', 'Inglese', 'Storia', 'Arte']
maschi = [30, 25, 35, 20, 15] # Numero di studenti maschi per materia
femmine = [20, 30, 25, 15, 10] # Numero di studentesse per materia
```



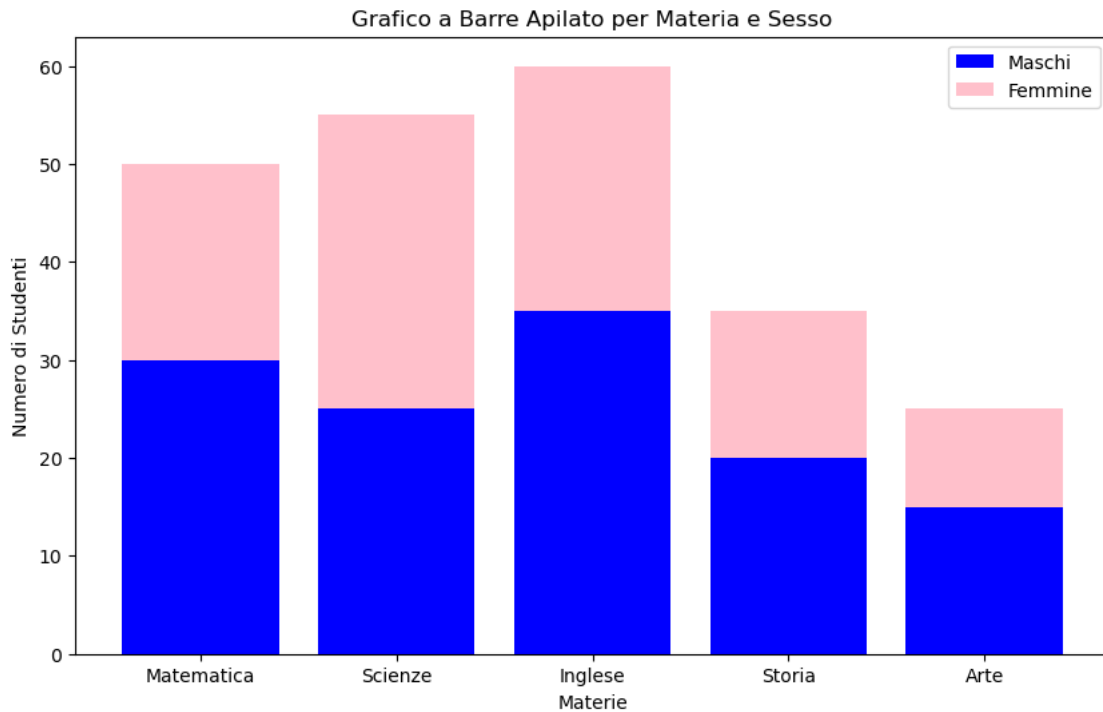
```

# Creare il grafico a barre apilato
plt.figure(figsize=(10, 6)) # Imposta le dimensioni del grafico
plt.bar(materie, maschi, label='Maschi', color='blue')
plt.bar(materie, femmine, label='Femmine', bottom=maschi, color='pink')

# Personalizzare il grafico
plt.title('Grafico a Barre Apilato per Materia e Sesso')
plt.xlabel('Materie')
plt.ylabel('Numero di Studenti')
plt.legend(loc='upper right')

# Mostra il grafico
plt.show()

```



1.16 Grafico a Linee delle Prestazioni per Annata

```

[20]: # Creazione di liste per rappresentare gli anni e le prestazioni dei tre gruppi
annata = ['Anno1', 'Anno2', 'Anno3']
gruppo1 = [90, 85, 88]
gruppo2 = [78, 92, 80]
gruppo3 = [85, 79, 91]

```

```

# Creazione di un grafico a linee per visualizzare le prestazioni nel tempo per
↳ciascun gruppo
plt.plot(annata, gruppo1, marker='o', label='Gruppo 1', linestyle='-',
↳color='blue')
plt.plot(annata, gruppo2, marker='s', label='Gruppo 2', linestyle='--',
↳color='green')
plt.plot(annata, gruppo3, marker='^', label='Gruppo 3', linestyle='-.',
↳color='red')

# Aggiunta di titolo e etichette agli assi
plt.title('Grafico a Linee delle Prestazioni per Annata')
plt.xlabel('Anno')
plt.ylabel('Prestazioni')

# Aggiunta della legenda in alto a sinistra per identificare i gruppi nel grafico
plt.legend(loc='upper left')

# Visualizzazione del grafico
plt.show()

```



1.17 Grafico a Barre Raggruppate delle Prestazioni dei Gruppi per Anno

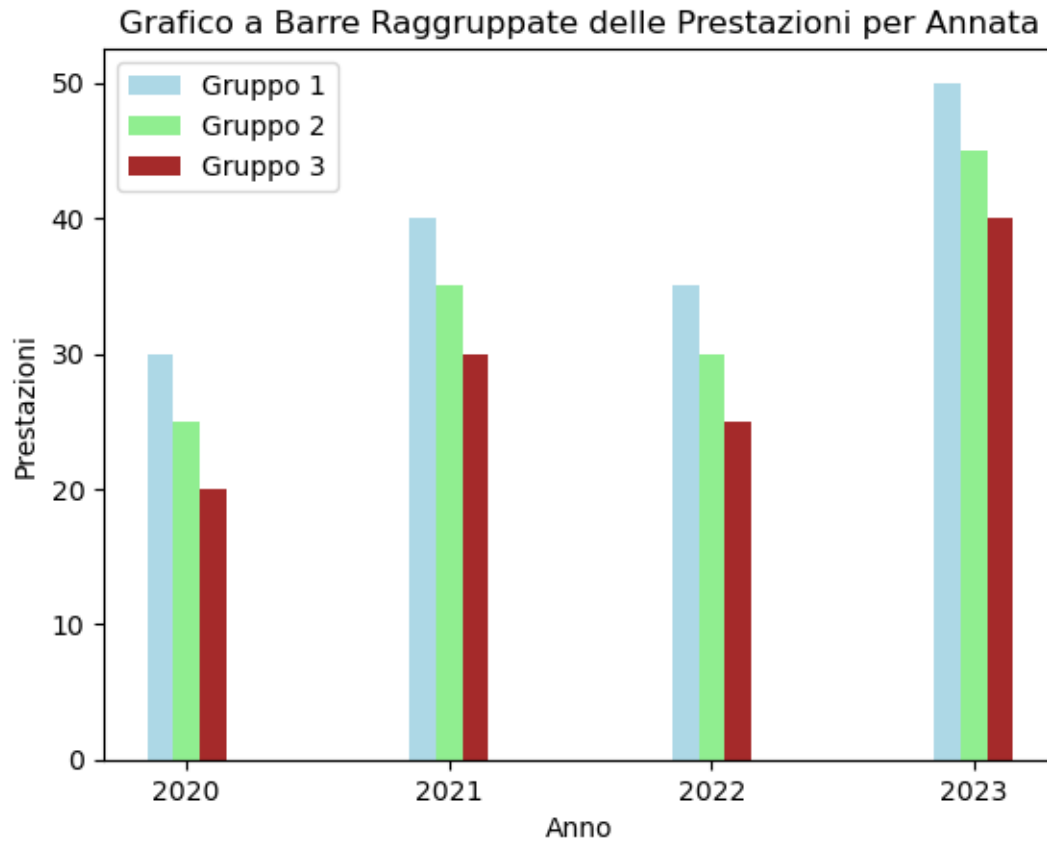
```
[21]: # Importazione delle librerie necessarie per il grafico
import matplotlib.pyplot as plt
import numpy as np

# Dati di esempio: annate e prestazioni dei tre gruppi
annata = ['2020', '2021', '2022', '2023']
gruppo1 = [30, 40, 35, 50]
gruppo2 = [25, 35, 30, 45]
gruppo3 = [20, 30, 25, 40]

# Definizione della larghezza delle barre e degli indici per la posizione delle
#   ↳ barre sul grafico
larghezza_barre = 0.1
indici = np.arange(len(annata)) #len(annata): Restituisce la lunghezza della
#   ↳ lista annata, cioè il numero di elementi presenti in essa. Nel tuo caso,
#   ↳ rappresenta il numero di anni. np.arange(len(annata)): Utilizza la funzione
#   ↳ arange di NumPy per creare un array di numeri interi da 0 a len(annata) - 1.
#   ↳ Questo array di numeri interi viene assegnato alla variabile indici. Quindi,
#   ↳ se annata contiene ['2020', '2021', '2022', '2023'], len(annata) sarà 4, e np.
#   ↳ arange(len(annata)) creerà l'array [0, 1, 2, 3]. Questi valori rappresentano
#   ↳ gli indici che vengono utilizzati per posizionare le barre nel grafico in modo
#   ↳ che siano distribuite uniformemente lungo l'asse x per ogni anno nel dataset
# Creazione di un grafico a barre raggruppate per visualizzare le prestazioni
#   ↳ dei tre gruppi per ogni anno
plt.bar(indici - larghezza_barre, gruppo1, width=larghezza_barre, label='Gruppo
#   ↳ 1', color='lightblue') #indici - larghezza_barre: Posiziona le barre del Gruppo
#   ↳ 1 a sinistra rispetto agli indici calcolati in precedenza, ottenendo così una
#   ↳ separazione tra i gruppi di barre.
plt.bar(indici, gruppo2, width=larghezza_barre, label='Gruppo 2',
#   ↳ color='lightgreen')
plt.bar(indici + larghezza_barre, gruppo3, width=larghezza_barre, label='Gruppo
#   ↳ 3', color='brown')

# Aggiunta di titolo, etichette degli assi e legenda
plt.title('Grafico a Barre Raggruppate delle Prestazioni per Annata')
plt.xlabel('Anno')
plt.ylabel('Prestazioni')
plt.xticks(indici, annata) # Etichette degli assi con gli anni
plt.legend(loc='upper left')

# Visualizzazione del grafico
plt.show()
```



```
[22]: indici = np.arange(len(annata))
```

```
[23]: indici - larghezza_barre
```

```
[23]: array([-0.1,  0.9,  1.9,  2.9])
```

1.18 Grafico a Barre Empilate delle Vendite Mensili per Prodotto

```
[1]: import matplotlib.pyplot as plt
import numpy as np

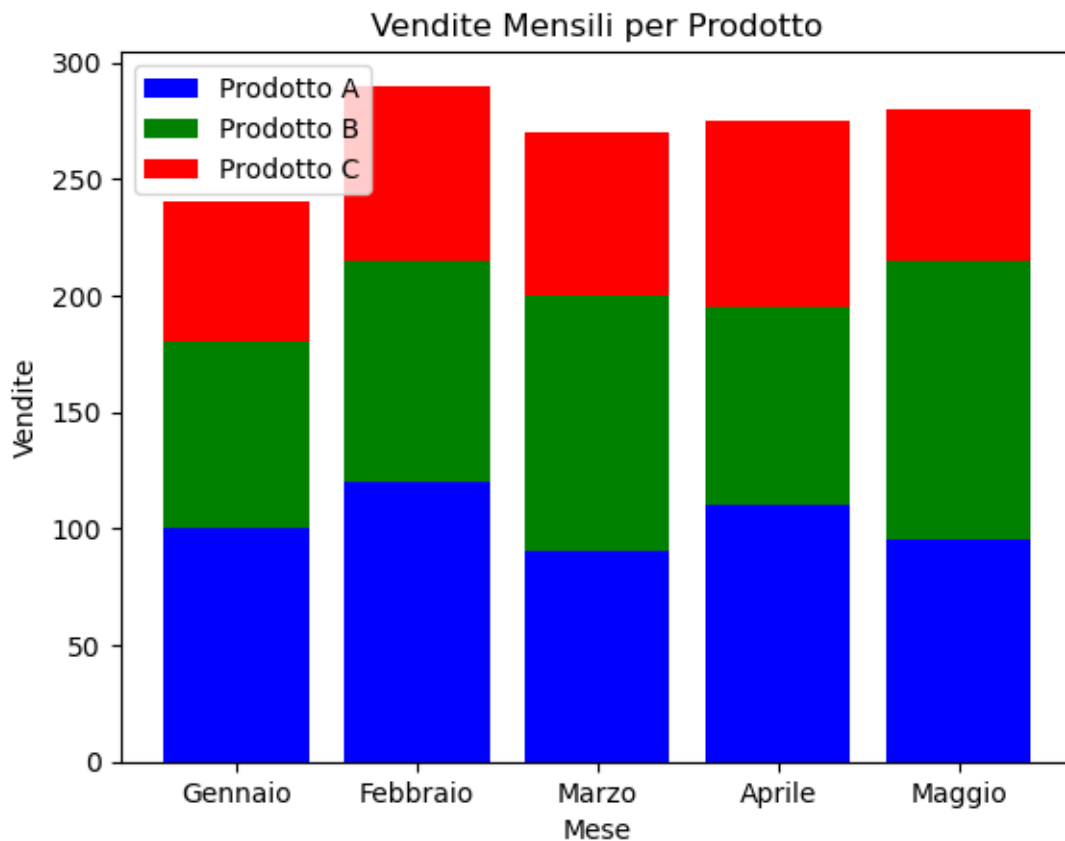
# Passo 2: Crea dati di esempio
mesi = ['Gennaio', 'Febbraio', 'Marzo', 'Aprile', 'Maggio']
vendite_prodotto_A = [100, 120, 90, 110, 95]
vendite_prodotto_B = [80, 95, 110, 85, 120]
vendite_prodotto_C = [60, 75, 70, 80, 65]

# Passo 3: Crea un grafico a barre empilate
```

```
plt.bar(mesi, vendite_prodotto_A, label='Prodotto A', color='blue') # bottom è
↳ un parametro della funzione plt.bar che determina da dove inizia la pila di
↳ barre empile
plt.bar(mesi, vendite_prodotto_B, label='Prodotto B', color='green',
↳ bottom=vendite_prodotto_A) # Le barre del "Prodotto B" iniziano dalla cima
↳ delle barre del "Prodotto A", quindi la pila si sviluppa sopra le barre del
↳ "Prodotto A".
plt.bar(mesi, vendite_prodotto_C, label='Prodotto C', color='red', bottom=np.
↳ array(vendite_prodotto_A) + np.array(vendite_prodotto_B))

# Passo 4: Personalizza il grafico
plt.title('Vendite Mensili per Prodotto')
plt.xlabel('Mese')
plt.ylabel('Vendite')
plt.legend(loc='upper left')

# Passo 5: Mostra il grafico risultante
plt.show()
```



1.19 Grafico a Barre Orizzontali della Popolazione delle Città Italiane

```
[31]: # Importa le librerie necessarie per il grafico
import matplotlib.pyplot as plt
import numpy as np

# Crea dati di esempio
città = ['Roma', 'Milano', 'Napoli', 'Torino', 'Firenze']
popolazione = [2870433, 1366180, 972198, 883767, 382258]

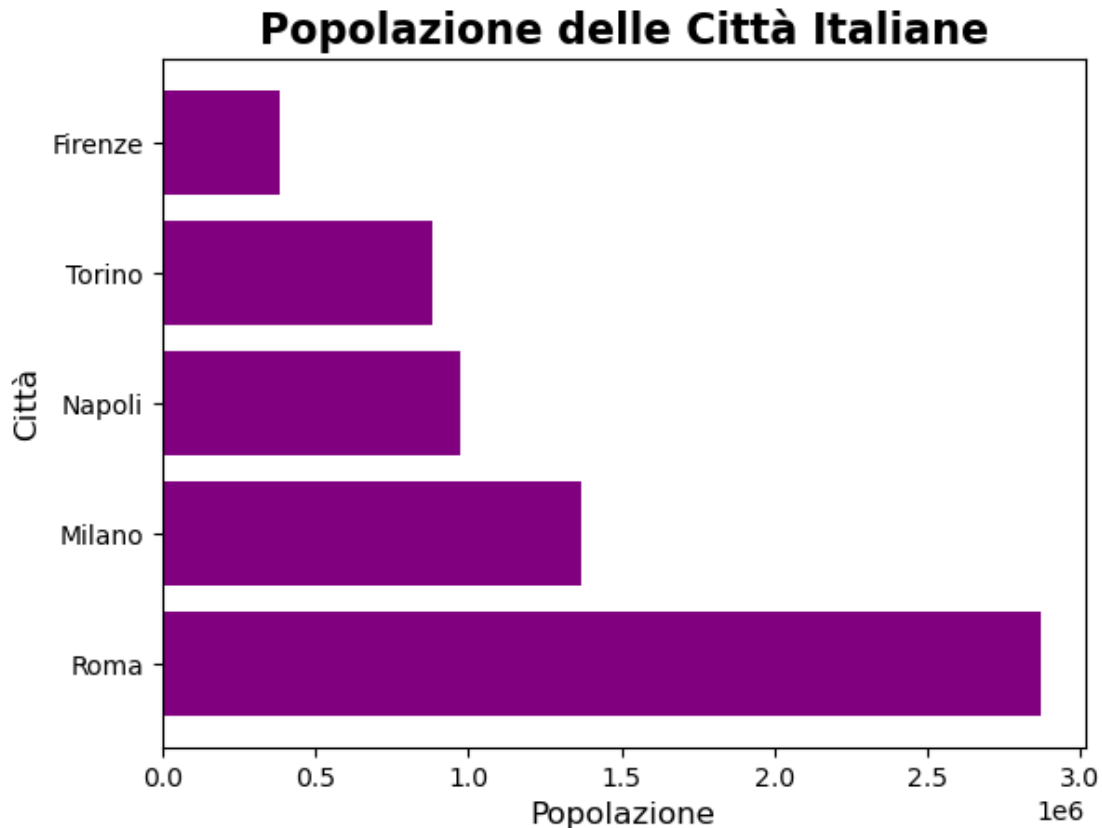
# Crea un grafico a barre orizzontali
plt.barh(città, popolazione, color='purple')

# Aggiungi un titolo al grafico
plt.title('Popolazione delle Città Italiane', fontsize=16, fontweight='bold')

# Etichetta l'asse x
plt.xlabel('Popolazione', fontsize=12)

# Etichetta l'asse y
plt.ylabel('Città', fontsize=12)

# Mostra il grafico risultante
plt.show()
```



1.20 Grafico a Linee dell'Andamento Mensile

```
[32]: # Importa le librerie necessarie per il grafico
import matplotlib.pyplot as plt
import numpy as np

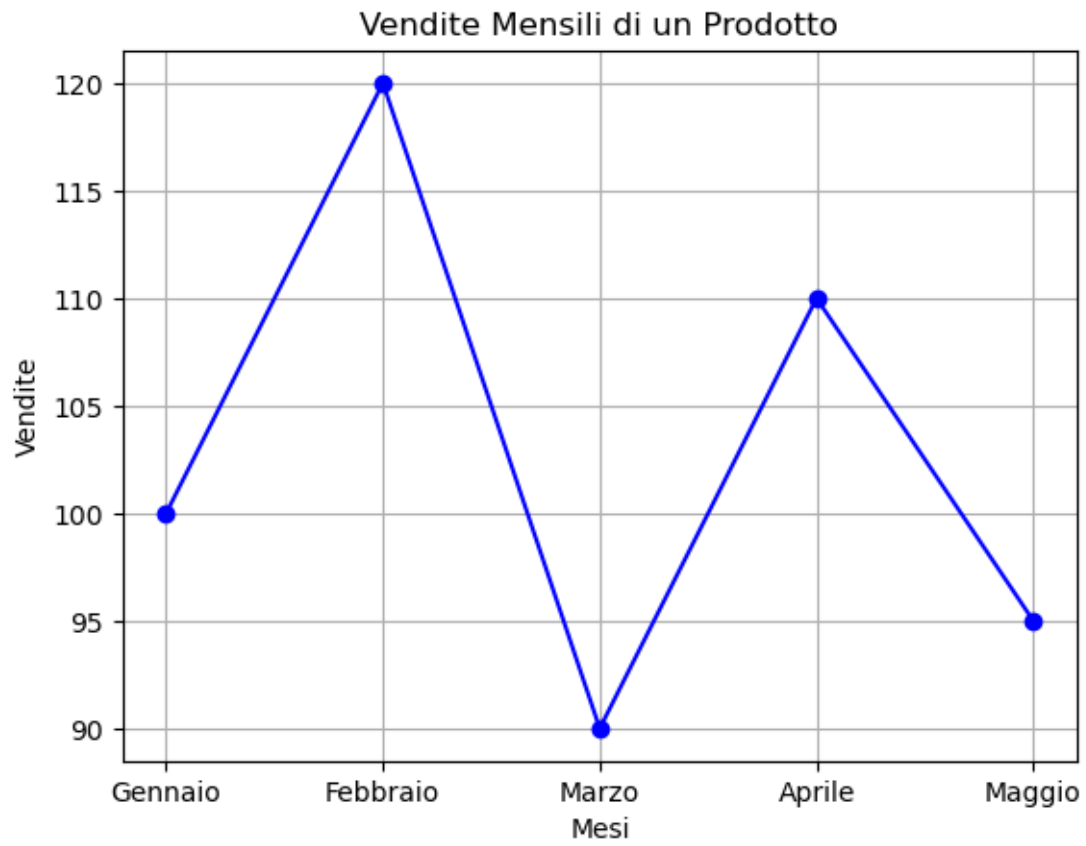
# Crea dati di esempio
mesi = ['Gennaio', 'Febbraio', 'Marzo', 'Aprile', 'Maggio']
vendite = [100, 120, 90, 110, 95]

# Crea un grafico a linee
plt.plot(mesi, vendite, marker='o', linestyle='-', color='blue')

# Personalizza il grafico
plt.title('Vendite Mensili di un Prodotto') # Titolo del grafico
plt.xlabel('Mesi') # Etichetta dell'asse x
plt.ylabel('Vendite') # Etichetta dell'asse y
plt.grid(True) # Mostra la griglia nel grafico

# Mostra il grafico risultante
```

```
plt.show()
```



1.21 Grafico a Torta della Distribuzione Percentuale delle Attività

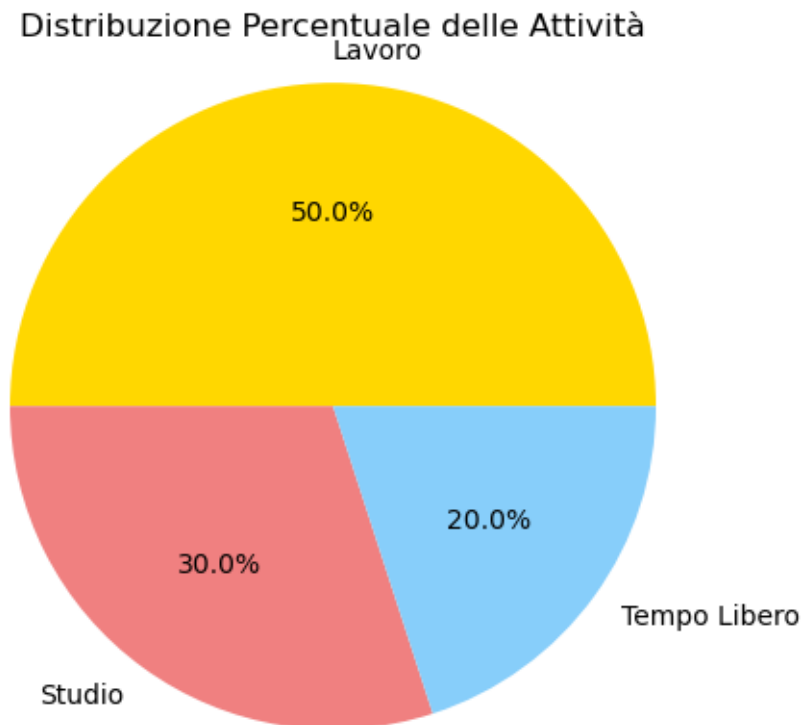

```
[6]: import matplotlib.pyplot as plt

# Passo 2: Crea dati di esempio
attività = ['Lavoro', 'Studio', 'Tempo Libero']
percentuali = [50, 30, 20]
colori = ['gold', 'lightcoral', 'lightskyblue']

# Passo 3: Crea un grafico a torta
plt.pie(percentuali, labels=attività, colors=colori, autopct='%1.1f%%')

# Passo 4: Personalizza il grafico
plt.title('Distribuzione Percentuale delle Attività') # Titolo del grafico
plt.axis('equal') # Rendi il grafico a torta circolare

# Passo 5: Mostra il grafico risultante
plt.show()
```



1.22 Diagramma a Dispersione del Confronto Punteggi di Matematica e Scienze

```
[35]: import matplotlib.pyplot as plt
import numpy as np

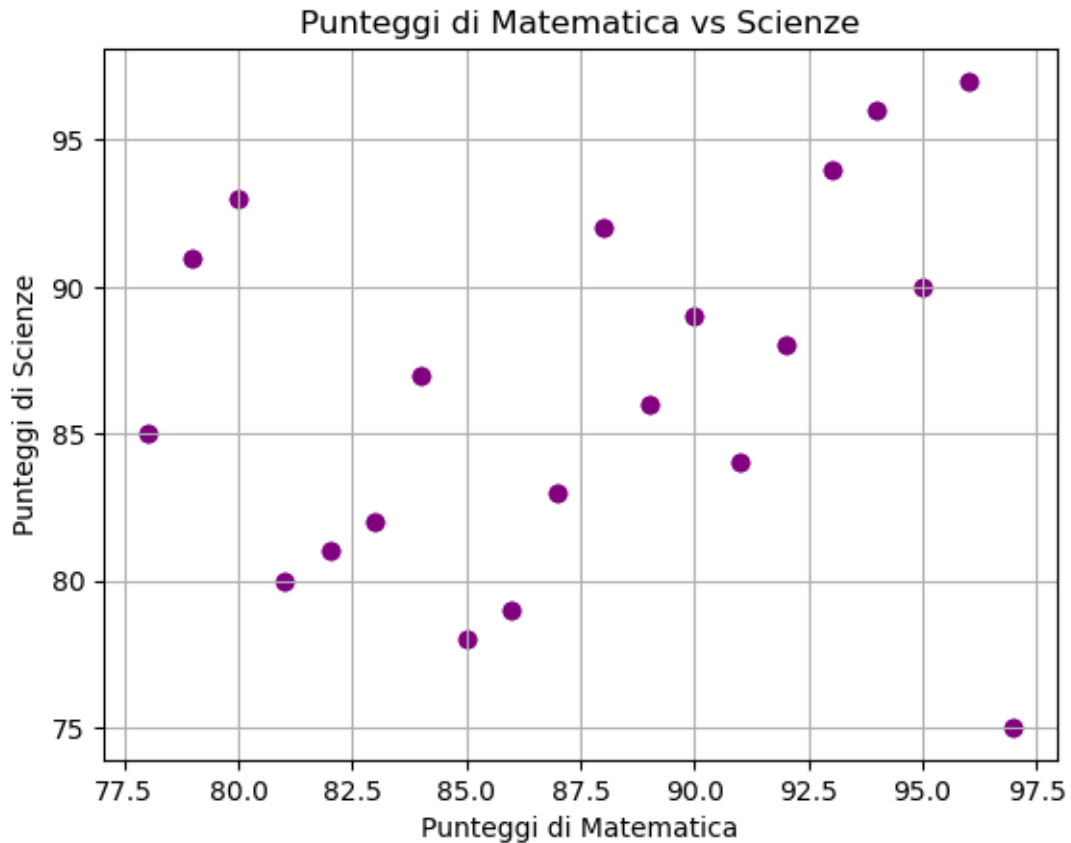
# Passo 2: Crea dati di esempio
punteggi_matematica = [85, 92, 78, 88, 95, 90, 89, 86, 79, 91, 84, 87, 83, 82,
↪81, 80, 93, 94, 96, 97, 85, 92, 78, 88, 95, 90, 89, 86, 79, 91, 84, 87, 83, 82,
↪81, 80, 93, 94, 96, 97]

punteggi_scienze = [78, 88, 85, 92, 90, 89, 86, 79, 91, 84, 87, 83, 82, 81, 80,
↪93, 94, 96, 97, 75, 78, 88, 85, 92, 90, 89, 86, 79, 91, 84, 87, 83, 82, 81, 80,
↪93, 94, 96, 97, 75]

# Passo 3: Crea un grafico a dispersione
plt.scatter(punteggi_matematica, punteggi_scienze, color='purple', marker='o')

# Passo 4: Personalizza il grafico
plt.title('Punteggi di Matematica vs Scienze')
plt.xlabel('Punteggi di Matematica')
plt.ylabel('Punteggi di Scienze')
plt.grid(True)

# Passo 5: Mostra il grafico risultante
plt.show()
```



1.23 Grafico a Dispersione del Confronto Punteggi di Matematica e Scienze

```
[36]: import random

punteggi_matematica = []
# Set a length of the list to 10
for i in range(0, 50):
    punteggi_matematica.append(random.randint(70, 100))

punteggi_scienze = []
# Set a length of the list to 10
for i in range(0, 50):
    # any random numbers from 0 to 1000
    punteggi_scienze.append(random.randint(70, 100))
```

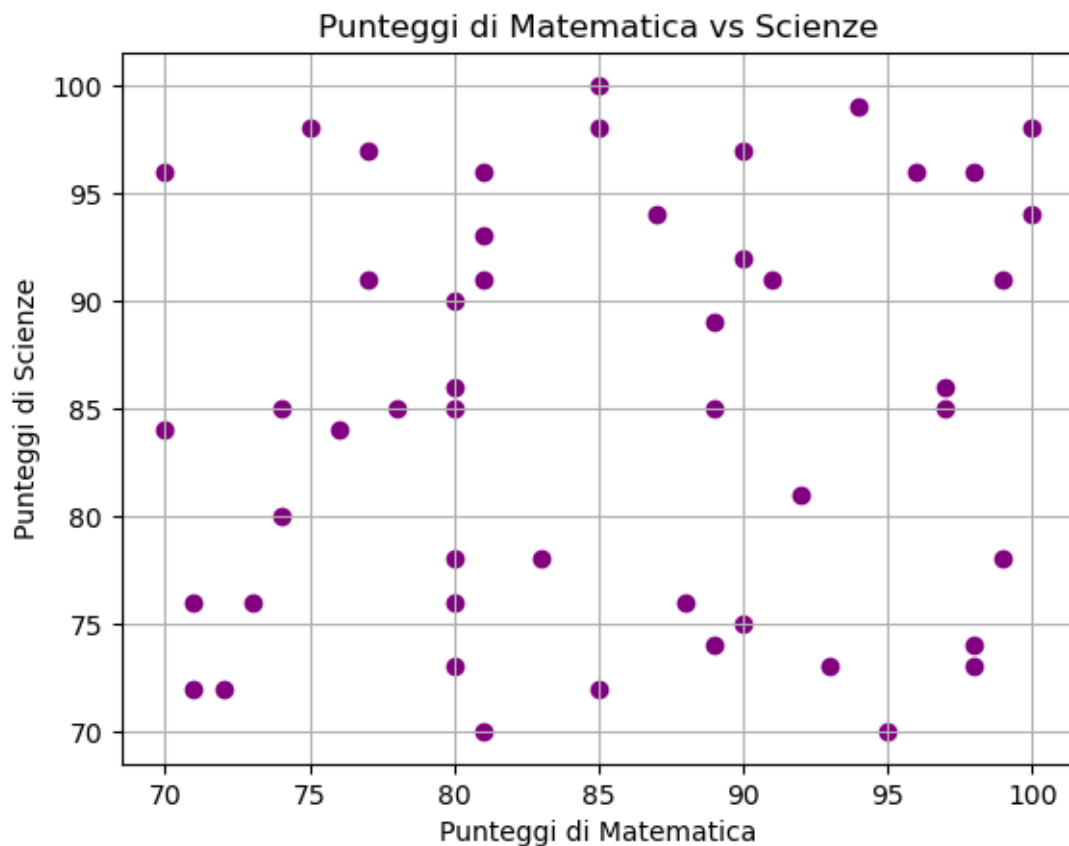
```

# Passo 3: Crea un grafico a dispersione
plt.scatter(punteggi_matematica, punteggi_scienze, color='purple', marker='o')

# Passo 4: Personalizza il grafico
plt.title('Punteggi di Matematica vs Scienze')
plt.xlabel('Punteggi di Matematica')
plt.ylabel('Punteggi di Scienze')
plt.grid(True)

# Passo 5: Mostra il grafico risultante
plt.show()

```



1.24 Grafico a Linee del Andamento delle Vendite Mensili

```

[8]: import matplotlib.pyplot as plt

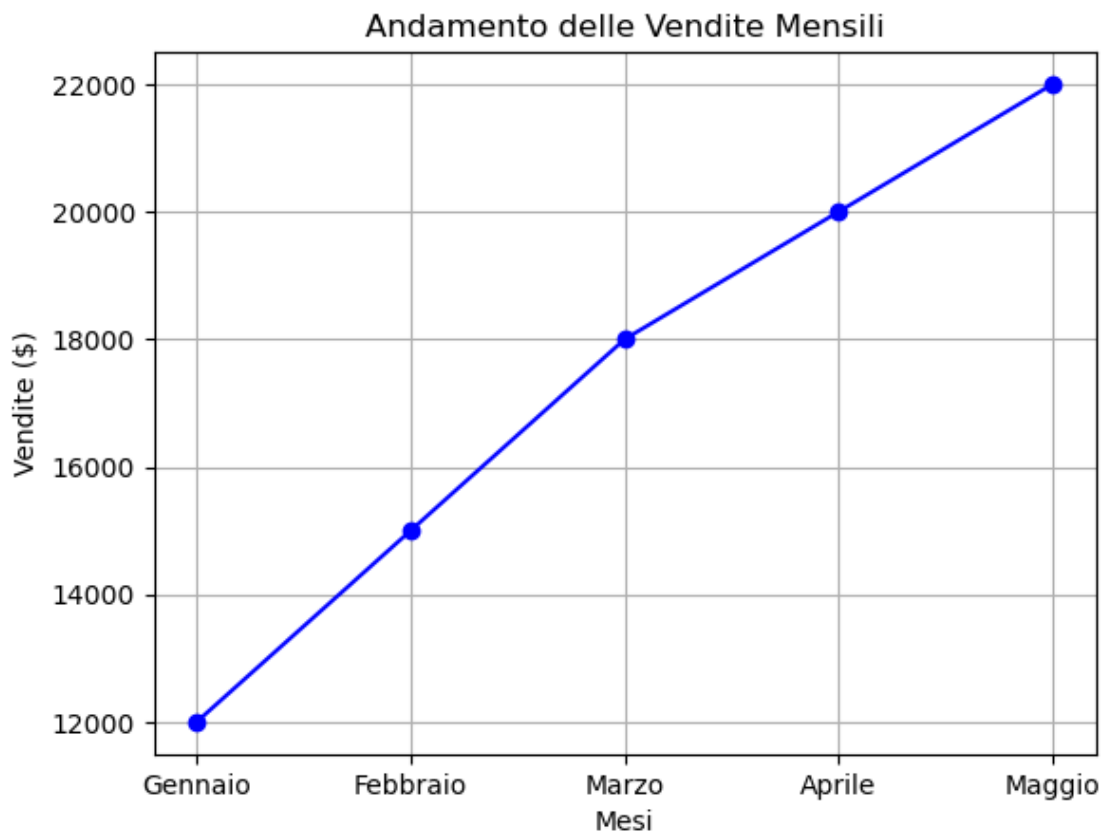
# Dati di esempio: mesi e corrispondenti vendite
mesi = ['Gennaio', 'Febbraio', 'Marzo', 'Aprile', 'Maggio']
vendite = [12000, 15000, 18000, 20000, 22000]

```

```
# Crea un grafico a linee
plt.plot(mesi, vendite, marker='o', linestyle='-', color='blue')

# Personalizza il grafico
plt.title('Andamento delle Vendite Mensili') # Titolo del grafico
plt.xlabel('Mesi') # Etichetta dell'asse x
plt.ylabel('Vendite ($)') # Etichetta dell'asse y
plt.grid(True) # Mostra la griglia nel grafico

# Mostra il grafico risultante
plt.show()
```



1.25 Grafico a Barre del Confronto delle Spese Annuali tra Anno 1 e Anno 2

```
[10]: import matplotlib.pyplot as plt
import numpy as np

# Dati di esempio: categorie e spese per due anni
categorie = ['Alimentari', 'Trasporti', 'Intrattenimento']
spese_anno_1 = [4000, 2500, 1500]
```

```

spese_anno_2 = [4200, 2700, 1600]

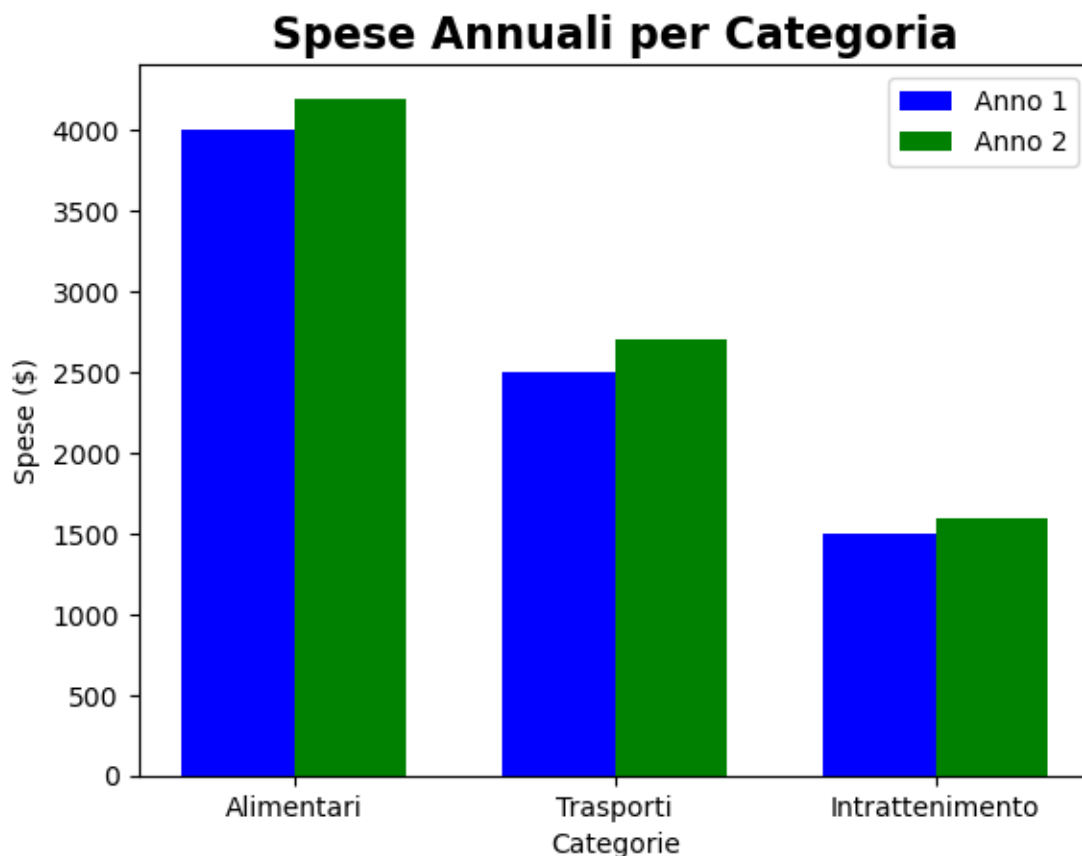
# Impostazioni per la larghezza delle barre e gli indici delle categorie
larghezza_barre = 0.35
indici = np.arange(len(categorie))

# Crea un grafico a barre per le spese annuali
plt.bar(indici - larghezza_barre/2, spese_anno_1, width=larghezza_barre,
        label='Anno 1', color='blue')
plt.bar(indici + larghezza_barre/2, spese_anno_2, width=larghezza_barre,
        label='Anno 2', color='green')

# Personalizza il grafico
plt.title('Spese Annuali per Categoria', fontsize=16, fontweight='bold') #
        ↳ Titolo del grafico
plt.xlabel('Categorie') # Etichetta dell'asse x
plt.ylabel('Spese ($)') # Etichetta dell'asse y
plt.xticks(indici, categorie) # Etichette per le categorie sull'asse x
plt.legend(loc='upper right') # Aggiunta della legenda in alto a destra

# Mostra il grafico risultante
plt.show()

```



1.26 Grafico a Dispersione del Confronto Punteggi di Matematica e Scienze

```
[15]: import matplotlib.pyplot as plt
import numpy as np

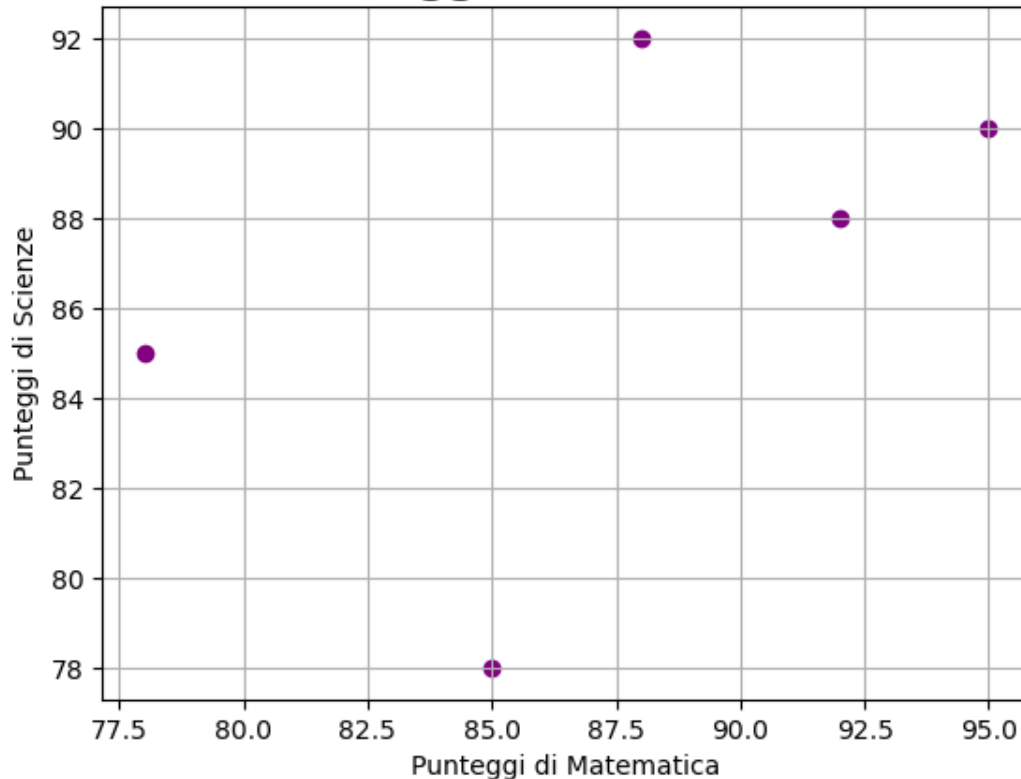
# Dati di esempio: punteggi di Matematica e Scienze per un gruppo di studenti
punteggi_matematica = [85, 92, 78, 88, 95]
punteggi_scienze = [78, 88, 85, 92, 90]

# Crea un grafico a dispersione
plt.scatter(punteggi_matematica, punteggi_scienze, color='purple', marker='o')

# Personalizza il grafico
plt.title('Confronto Punteggi di Matematica vs Scienze', fontsize=16,
↪fontweight='bold') # Titolo del grafico
plt.xlabel('Punteggi di Matematica') # Etichetta dell'asse x
plt.ylabel('Punteggi di Scienze') # Etichetta dell'asse y
plt.grid(True) # Mostra la griglia nel grafico

# Mostra il grafico risultante
plt.show()
```

Confronto Punteggi di Matematica vs Scienze



1.27 Grafico a Linee della Variazione della Temperatura Annuale Media (2010-2015)

```
[17]: import matplotlib.pyplot as plt

# Dati di esempio: anni e temperatura media annuale
anni = [2010, 2011, 2012, 2013, 2014, 2015]
temperatura_media = [15, 16, 16.5, 17, 16.2, 15.8]

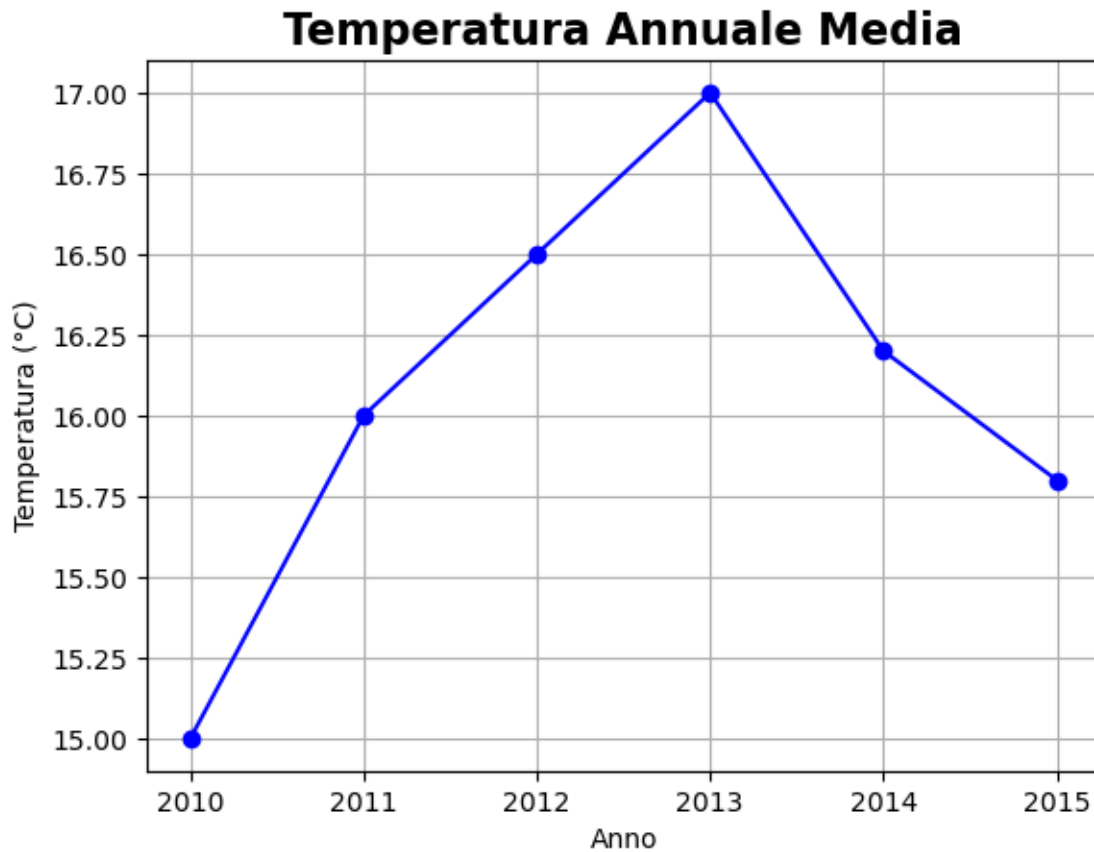
# Crea un grafico a linee
plt.plot(anni, temperatura_media, marker='o', linestyle='-', color='blue')

# Personalizza il grafico
plt.title('Temperatura Annuale Media', fontsize=16, fontweight='bold') # Titolo
# del grafico
plt.xlabel('Anno') # Etichetta dell'asse x
plt.ylabel('Temperatura (°C)') # Etichetta dell'asse y
plt.grid(True) # Mostra la griglia nel grafico

# Mostra il grafico risultante
```



```
plt.show()
```



1.28 Grafico a Barre della Distribuzione della Popolazione per Età e Genere

```
[18]: import matplotlib.pyplot as plt
import numpy as np

# Dati di esempio: età, popolazione maschile e popolazione femminile
età = ['0-18', '19-35', '36-50', '51-65', '66+']
popolazione_maschile = [1000, 2500, 1800, 1200, 800]
popolazione_femminile = [950, 2400, 1700, 1100, 850]

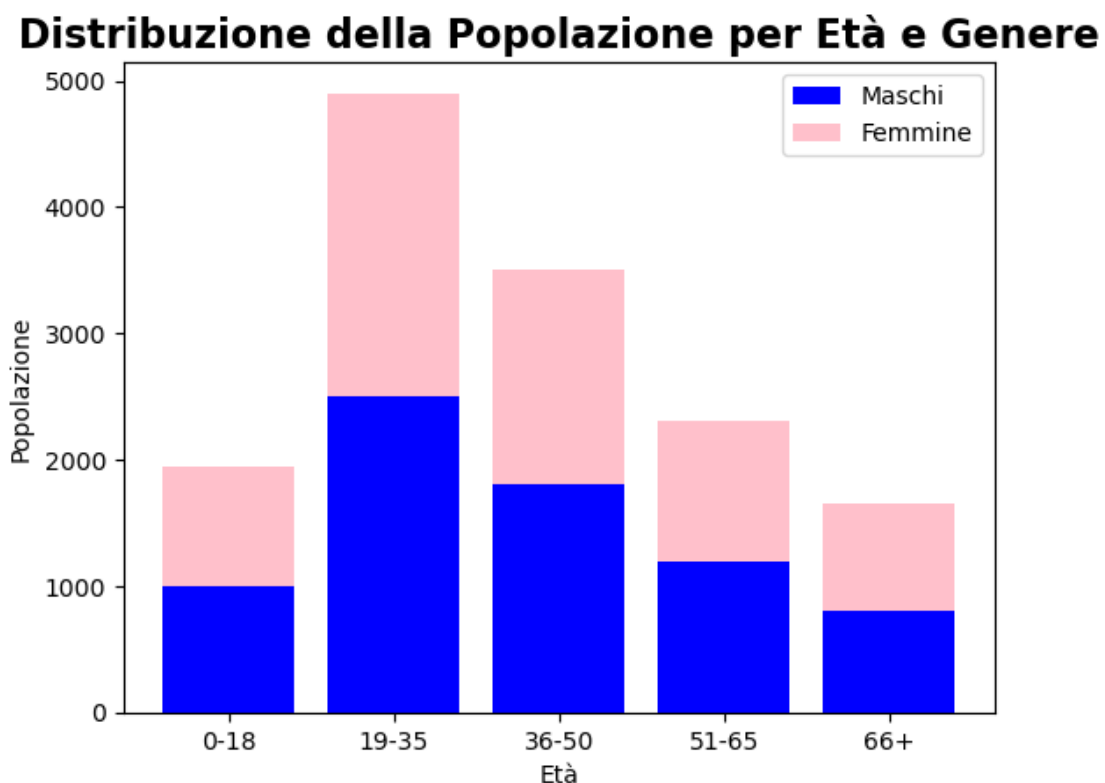
# Impostazioni per gli indici delle età
indici = np.arange(len(età))

# Crea un grafico a barre empilate per la distribuzione della popolazione per
# →età e genere
plt.bar(età, popolazione_maschile, label='Maschi', color='blue')
```

```
plt.bar(età, popolazione_femminile, label='Femmine',
        ↳bottom=popolazione_maschile, color='pink')

# Personalizza il grafico
plt.title('Distribuzione della Popolazione per Età e Genere', fontsize=16,
        ↳fontweight='bold') # Titolo del grafico
plt.xlabel('Età') # Etichetta dell'asse x
plt.ylabel('Popolazione') # Etichetta dell'asse y
plt.legend(loc='upper right') # Aggiunta della legenda in alto a destra

# Mostra il grafico risultante
plt.show()
```



1.29 Piramide Demografica per Genere e per Età

```
[13]: import matplotlib.pyplot as plt
import numpy as np

# Dati di esempio: fasce d'età, popolazione maschile e popolazione femminile
fasce_eta = ['0-4', '5-9', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39',
        ↳'40-44', '45-49']
```

```

popolazione_maschile = [1550000, 1600000, 1650000, 1700000, 1750000, 1780000,
↳1755000, 1720000, 1675000, 1600000]
popolazione_femminile = [1480000, 1550000, 1605000, 1650000, 1705000, 1750000,
↳1760000, 1725000, 1680000, 1595000]

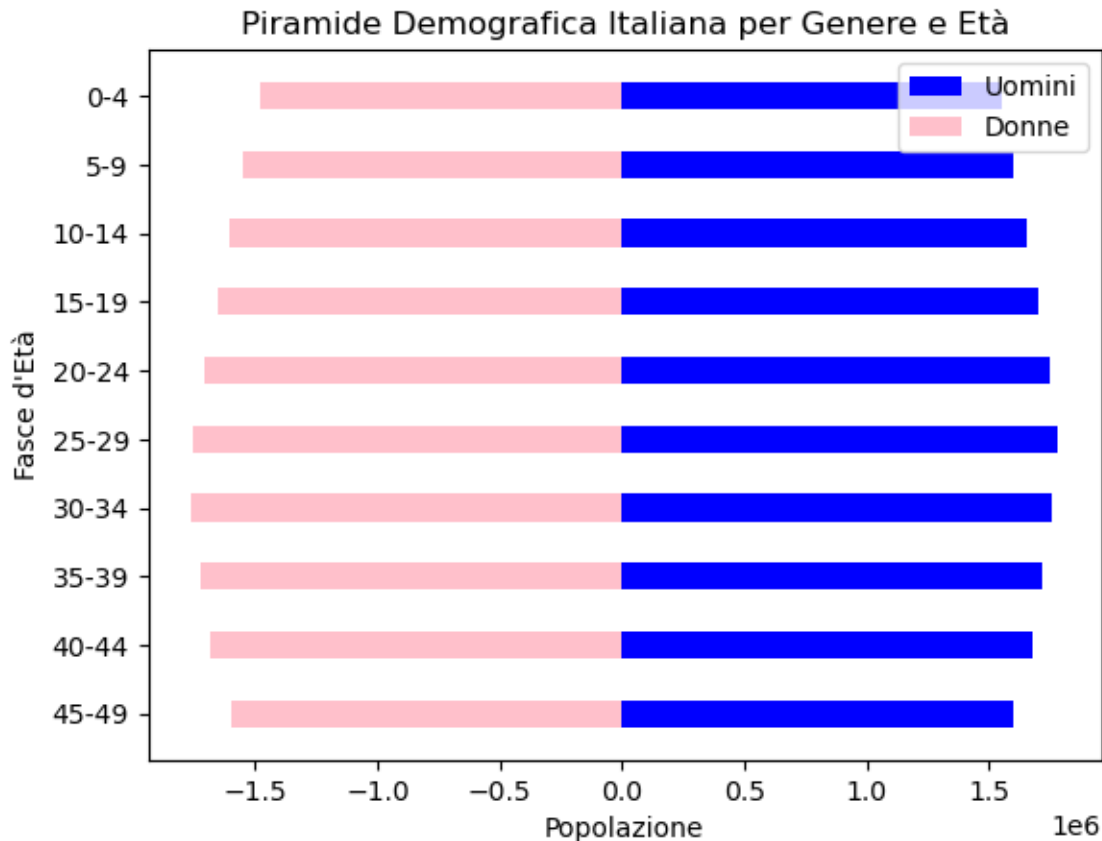
# Impostazioni per gli indici delle fasce d'età e la larghezza delle barre
indici = np.arange(len(fasce_eta))
larghezza_barre = 0.4

# Creazione della piramide demografica
fig, ax1 = plt.subplots()

ax1.barh(indici, popolazione_maschile, height=larghezza_barre, label='Uomini',
↳color='blue')
ax1.barh(indici, np.negative(popolazione_femminile), height=larghezza_barre,
↳label='Donne', color='pink')

# Personalizzazione del grafico
ax1.set_xlabel('Popolazione')
ax1.set_ylabel('Fasce d\'Età')
ax1.set_title('Piramide Demografica Italiana per Genere e Età')
ax1.set_yticks(indici)
ax1.set_yticklabels(fasce_eta)
ax1.legend(loc='upper right')
ax1.invert_yaxis() # Inverti l'asse y per rendere la piramide demografica
# Mostra il grafico risultante
plt.show()

```



1.30 Piramide Demografica Italiana per Genere e Età (Dati Fittizi)

```
[46]: import matplotlib.pyplot as plt
import numpy as np

# Fasce d'età della popolazione italiana
fasce_eta = ['0-4', '5-9', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '40-44', '45-49',
            '50-54', '55-59', '60-64', '65-69', '70-74', '75-79', '80-84', '85-89', '90-94', '95-99', '100+']

# Popolazione totale italiana (in milioni)
popolazione_totale = 60

# Stima realistica della percentuale di popolazione maschile per ogni fascia di età (dati ipotetici)
percentuali_maschili = [49.5, 49.0, 48.5, 48.0, 47.5, 47.0, 46.5, 46.0, 45.5, 45.0,
                        44.5, 44.0, 43.5, 43.0, 42.5, 42.0, 41.5, 41.0, 40.5, 40.0, 39.5, 39.0]
```

```

44.5, 44.0, 43.5, 43.0, 42.5, 42.0, 41.5, 41.0, 40.5, 40.
→0, 30.0]

popolazione_maschile = []
popolazione_femminile = []

# Calcolo della popolazione maschile per ogni fascia di età
for percentuale in percentuali_maschili:
    popolazione_maschile.append(round((percentuale / 100) * (popolazione_totale_
→* 1000000)))

# Calcolo della popolazione femminile per ogni fascia di età
for maschile in popolazione_maschile:
    popolazione_femminile = [(popolazione_totale * 1000000) - maschile]

# Impostazioni per gli indici delle fasce d'età e la larghezza delle barre
indici = np.arange(len(fasce_eta))
larghezza_barre = 0.4

# Creazione della piramide demografica
fig, ax1 = plt.subplots()

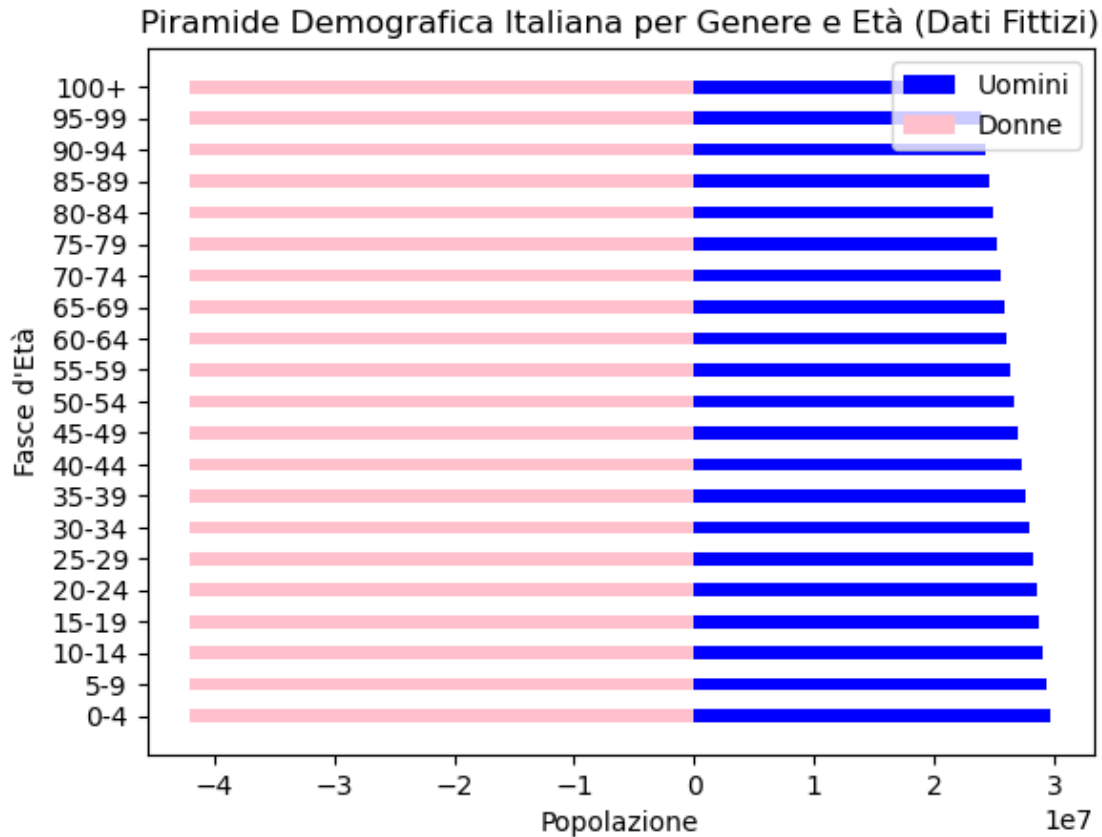
# Disegna le barre orizzontali per la popolazione maschile (barre blu)
ax1.barh(indici, popolazione_maschile, height=larghezza_barre, label='Uomini',
→color='blue')

# Disegna le barre orizzontali per la popolazione femminile (barre rosa)
ax1.barh(indici, [-x for x in popolazione_femminile], height=larghezza_barre,
→label='Donne', color='pink')

# Personalizzazione del grafico
ax1.set_xlabel('Popolazione')
ax1.set_ylabel('Fasce d\'Età')
ax1.set_title('Piramide Demografica Italiana per Genere e Età (Dati Fittizi)')
ax1.set_yticks(indici)
ax1.set_yticklabels(fasce_eta)
ax1.legend(loc='upper right')
#ax1.invert_yaxis() # Inverti l'asse y per rendere la piramide demografica

# Mostra il grafico risultante
plt.show()

```



```
[136]: import numpy as np
import matplotlib.pyplot as plt

# Esempio di dati (da sostituire con dati reali)
anni = [2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]
marchi_auto = ['Toyota', 'Ford', 'Chevrolet', 'Honda', 'Nissan', 'Volkswagen', '
↳ 'Hyundai', 'Mercedes', 'BMW', 'Audi']
vendite_annuali = np.random.randint(50000, 300000, size=(len(marchi_auto),
↳ len(anni)))#size=(len(marchi_auto), len(anni)) indica che stiamo generando una
↳ matrice con il numero di righe pari al numero di marchi auto e il numero di
↳ colonne pari al numero di anni. Questa matrice è quindi utilizzata per
↳ simulare le vendite annuali per ogni marchio auto negli anni specificati.

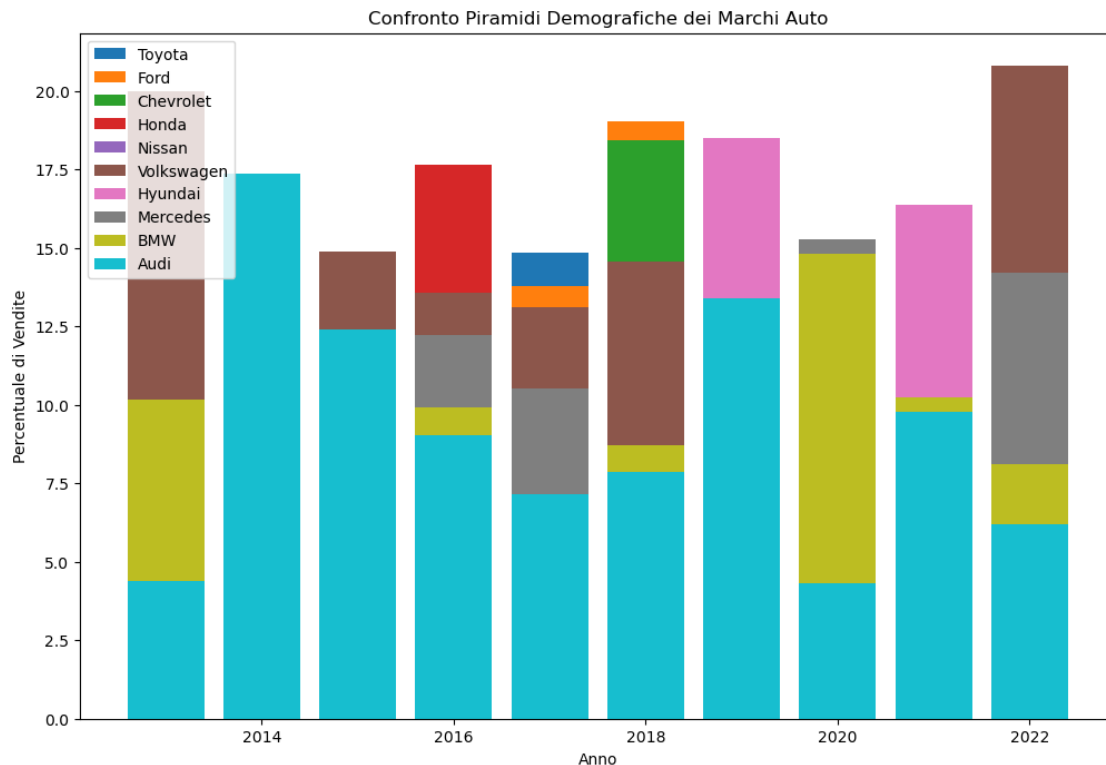
# Calcola le vendite totali per ogni anno
vendite_totali_annuali = np.sum(vendite_annuali, axis=0)#somma delle vendite per
↳ ogni anno.

# Creazione di una piramide demografica sovrapposta per ogni marchio
```

```
fig, ax = plt.subplots(figsize=(12, 8)) #La funzione plt.subplots() in Matplotlib
    ↳ viene utilizzata per creare una figura e un set di assi in una singola
    ↳ chiamata di funzione. Restituisce un oggetto figura e un array di oggetti assi.

for i, marchio in enumerate(marchi_auto):
    percentuali_per_anno = vendite_annuali[i, :] / vendite_totali_annuali * 100#.
    ↳ La riga i corrisponde al marchio specifico di interesse, e il simbolo : è uno
    ↳ slicing che indica di selezionare tutte le colonne corrispondenti a ciascun
    ↳ anno.
    ax.bar(anni, percentuali_per_anno, label=marchio)

ax.set_xlabel('Anno')
ax.set_ylabel('Percentuale di Vendite')
ax.set_title('Confronto Piramidi Demografiche dei Marchi Auto')
ax.legend()
plt.show()
```



[]: