

```
In [1]: 1 # Importa Librerie
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import matplotlib.patches as mpatches
5 import numpy as np
6 from math import pi
```

```
In [2]: 1 # Dati dei voti [2021, 2022]
2 voti = {
3     'Pier': [8, 4],
4     'Clara': [5, 7],
5     'SdG': [4, 6.75],
6     'Belmy': [2, 5],
7     'Sofia': [6.5, 7.75],
8     'Valeria': [7.5, 7.5],
9     'Clarke': [7, 8.5],
10    'Simo': [7, 8.5],
11    'Dudu': [3.5, 5.5],
12    'Jacopo': [5, 6.5],
13    'Ludona': [2, 6],
14    'Luisa': [8, 8],
15    'Sere': [7, 8],
16    'Michele': [8, 7.25],
17    'Gaia': [8, 9]
18 }
```

```
In [3]: 1 # Crea il DataFrame
2 df = pd.DataFrame(voti, index=['2021', '2022'])
3
4 # Mostra il DataFrame
5 print(df)
```

	Pier	Clara	SdG	Belmy	Sofia	Valeria	Clarke	Simo	Dudu	Jacopo	\
2021	8	5	4.00	2	6.50	7.5	7.0	7.0	3.5	5.0	
2022	4	7	6.75	5	7.75	7.5	8.5	8.5	5.5	6.5	

	Ludona	Luisa	Sere	Michele	Gaia
2021	2	8	7	8.00	8
2022	6	8	8	7.25	9

```
In [4]: 1 # Trasponi il DataFrame
2 df_t = df.T
```

```
In [5]: 1 # Media
2 media = df_t.mean()
3 print(media)
```

```
2021    5.900000
2022    7.016667
dtype: float64
```

```
In [6]: 1 # Moda
2 moda = df_t.mode()
3 print(moda)
```

```
2021  2022
0    8.0    8.0
1    NaN    8.5
```

```
In [7]: 1 # Mediana
2 mediana = df_t.median()
3 print(mediana)
```

```
2021    7.00
2022    7.25
dtype: float64
```

```
In [8]: 1 # Varianza
2 varianza = df_t.var()
3 print(varianza)
```

```
2021    4.685714
2022    1.995238
dtype: float64
```

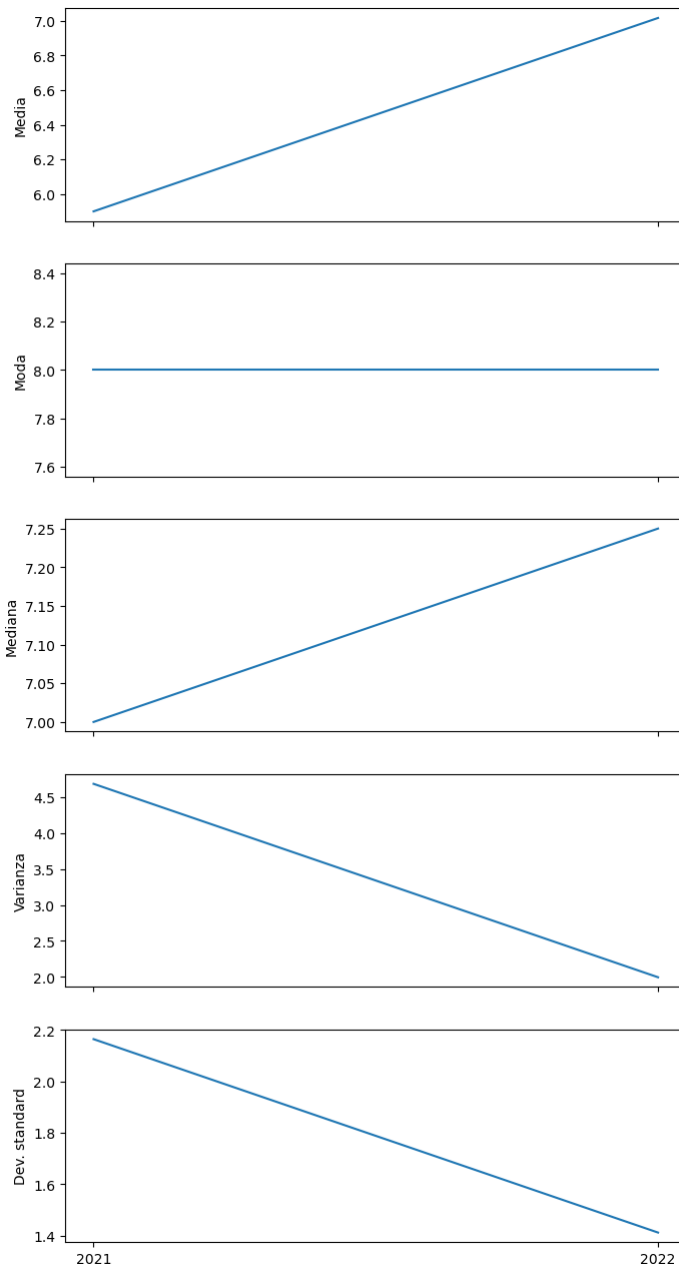
```
In [9]: 1 # Deviazione standard
2 dev_std = df_t.std()
3 print(dev_std)
```

```
2021    2.164651
2022    1.412529
dtype: float64
```

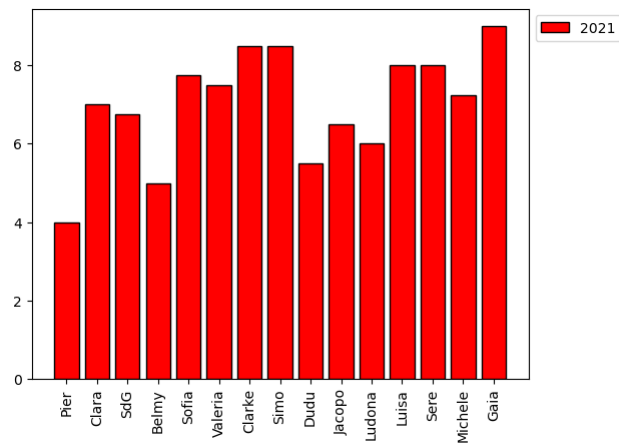
```

In [10]: 1 # Crea una figura con 5 sottografici
2 fig, ax = plt.subplots(5, 1, sharex=True, figsize=(8, 16))
3
4 # Crea il grafico a linee per la media
5 ax[0].plot(df_t.mean())
6 ax[0].set_ylabel('Media')
7
8 # Crea il grafico a linee per la moda
9 ax[1].plot(df_t.mode().iloc[0])
10 ax[1].set_ylabel('Moda')
11
12 # Crea il grafico a linee per la mediana
13 ax[2].plot(df_t.median())
14 ax[2].set_ylabel('Mediana')
15
16 # Crea il grafico a linee per la varianza
17 ax[3].plot(df_t.var())
18 ax[3].set_ylabel('Varianza')
19
20 # Crea il grafico a linee per la deviazione standard
21 ax[4].plot(df_t.std())
22 ax[4].set_ylabel('Dev. standard')
23
24 # Mostra la figura
25 plt.show()

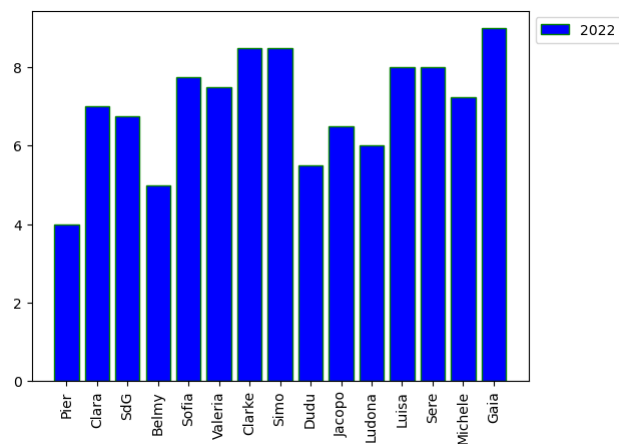
```



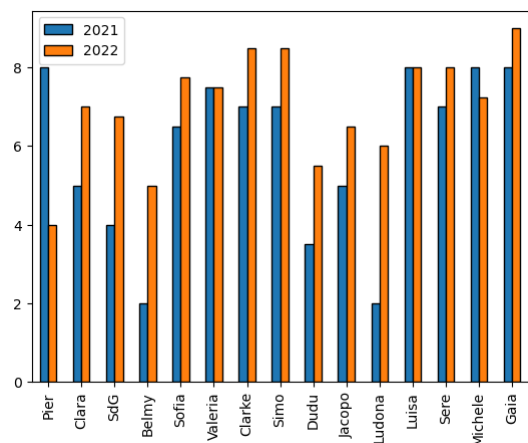
```
In [11]: 1 # Crea una lista con i nomi delle persone
2 nomipersona = list(voti.keys())
3
4 # Crea una lista con i voti del 2021
5 voti_2021 = [voti[nome][1] for nome in voti]
6
7 # Crea il grafico a barre
8 plt.bar(nomipersona, voti_2021, color='red', edgecolor='black')
9 plt.legend(['2021'], loc='upper left', bbox_to_anchor=(1,1))
10
11 # Ruota le etichette delle barre
12 plt.xticks(rotation=90)
13
14 plt.show()
```



```
In [12]: 1 # Crea una lista con i nomi delle persone
2 nomipersona = list(voti.keys())
3
4 # Crea una lista con i voti del 2022
5 voti_2022 = [voti[nome][2] for nome in voti]
6
7 # Crea il grafico a barre
8 plt.bar(nomipersona, voti_2022, color='blue', edgecolor='green')
9 plt.legend(['2022'], loc='upper left', bbox_to_anchor=(1,1))
10
11 # Ruota le etichette delle barre
12 plt.xticks(rotation=90)
13
14 plt.show()
```



```
In [13]: 1 # Grafico a barre
2 df_t.plot.bar(edgecolor='black')
3 plt.show()
```



```

In [14]: 1 def aaa(voti_anno, colore, bordo, anno):
2         frequenze = {}
3         for voto in voti_anno:
4             if voto in frequenze.keys():
5                 frequenze[voto] += 1
6             else:
7                 frequenze[voto] = 1
8         print(frequenze)
9
10        plt.figure(figsize=(16, 8))
11        plt.bar(frequenze.keys(), frequenze.values(), color=colore, edgecolor=bordo, width=0.2)
12        plt.xticks(list(frequenze.keys()))
13        plt.legend([anno], loc="upper left", bbox_to_anchor=(1,1))
14        plt.show()

```

```

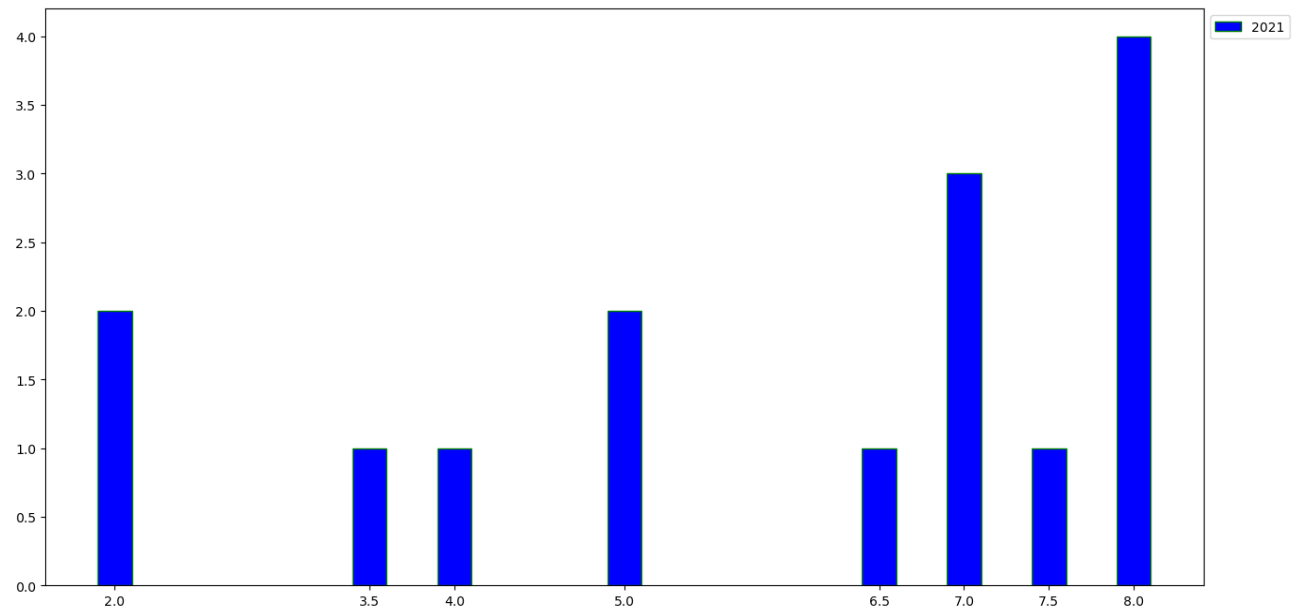
In [15]: 1 aaa([voti[nome][0] for nome in voti], 'blue', 'green', '2021')

```

```

{8: 4, 5: 2, 4: 1, 2: 2, 6.5: 1, 7.5: 1, 7: 3, 3.5: 1}

```



```

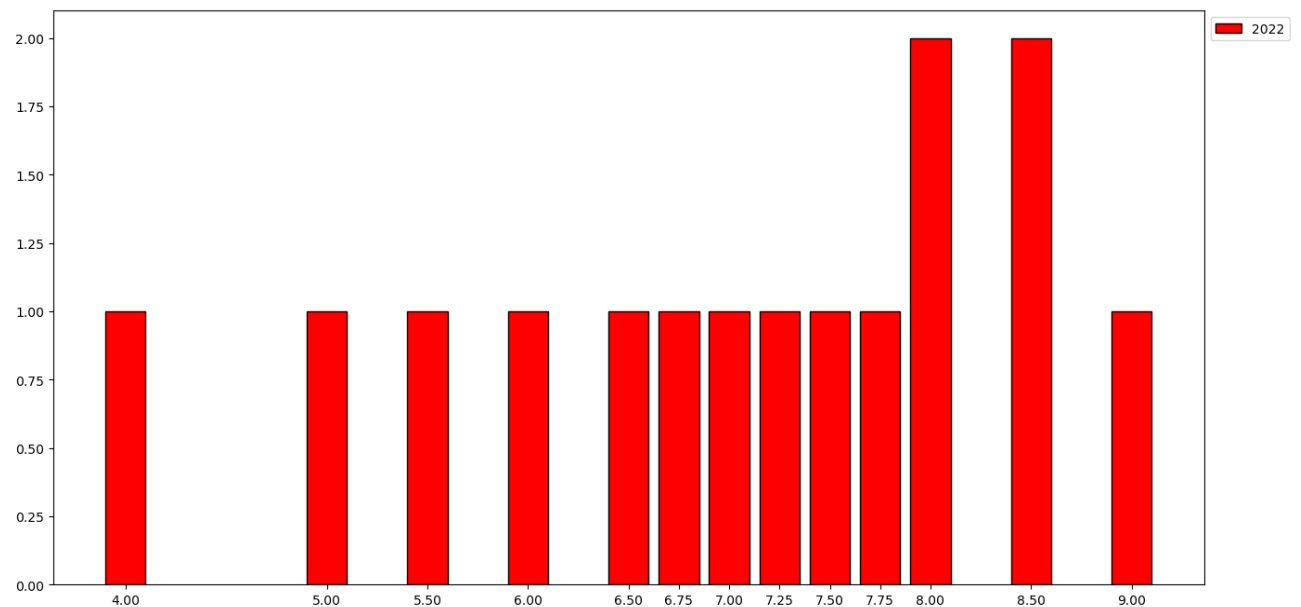
In [16]: 1 aaa([voti[nome][1] for nome in voti], 'red', 'black', '2022')

```

```

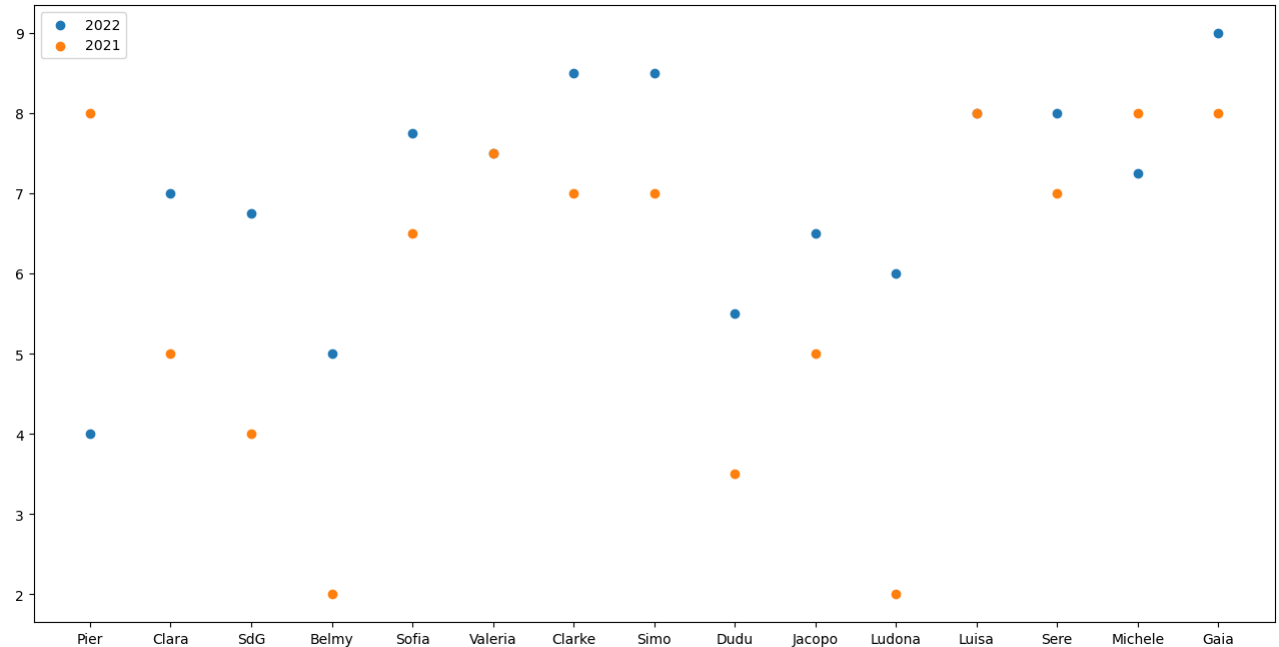
{4: 1, 7: 1, 6.75: 1, 5: 1, 7.75: 1, 7.5: 1, 8.5: 2, 5.5: 1, 6.5: 1, 6: 1, 8: 2, 7.25: 1, 9: 1}

```

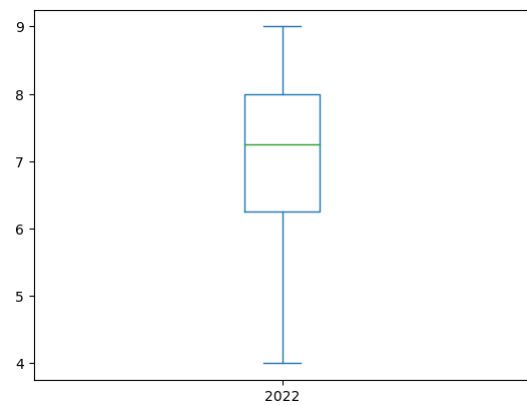


```
In [17]: 1 # Grafico a punti
2 # Crea La figura
3 fig = plt.figure()
4 plt.figure(figsize=(16, 8))
5
6 # Crea il grafico a punti
7 plt.scatter(df_t.index, df_t['2022'], label='2022')
8 plt.scatter(df_t.index, df_t['2021'], label='2021')
9
10 # Aggiungi Le etichette dell'asse x
11 plt.xticks(df_t.index)
12
13 # Aggiungi La Legenda
14 plt.legend()
15
16 # Mostra La figura
17 plt.show()
```

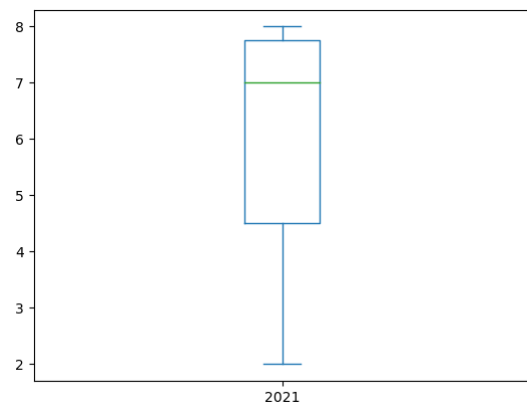
<Figure size 640x480 with 0 Axes>



```
In [18]: 1 # Box plot per il 2022
2 df_t['2022'].plot.box()
3 plt.show()
```



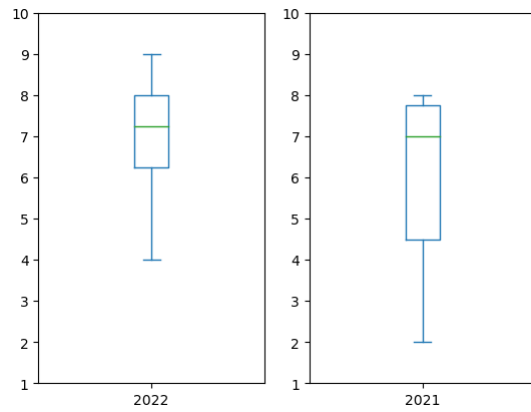
```
In [19]: 1 # Box plot per il 2021
2 df_t['2021'].plot.box()
3 plt.show()
```



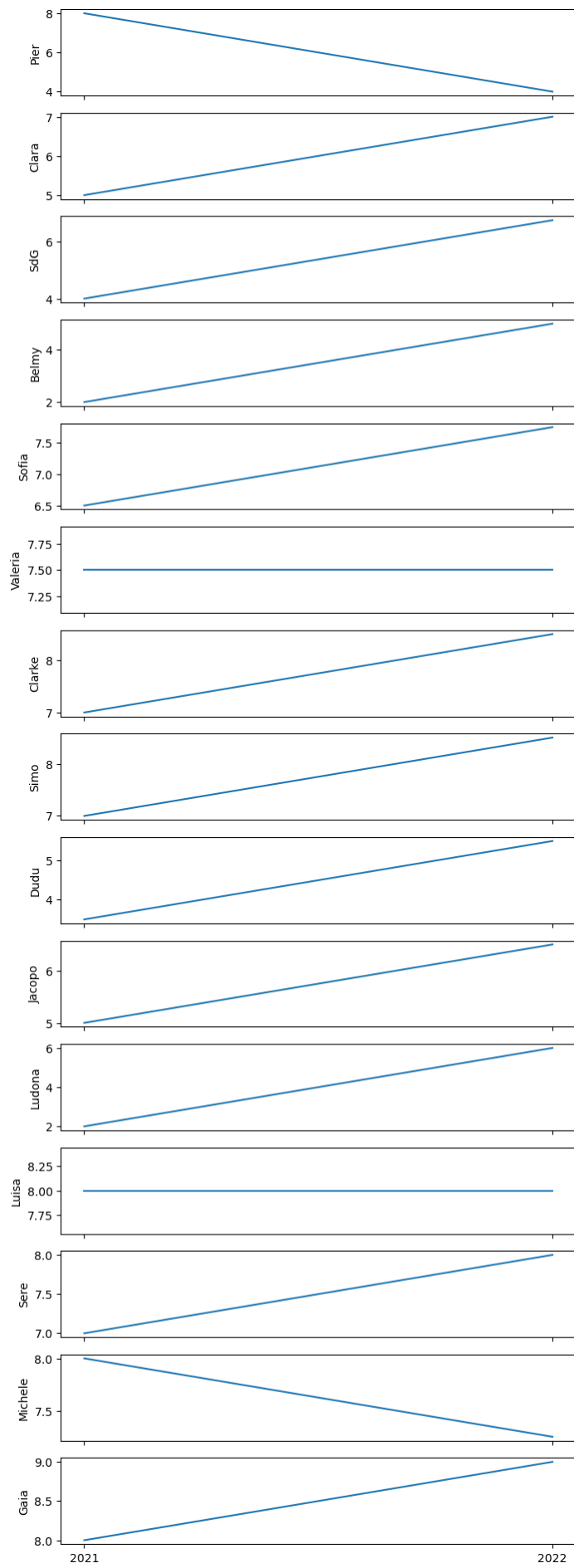
```

In [20]: 1 # Crea una figura con 2 sottografici
2 fig, ax = plt.subplots(1, 2)
3
4 # Box plot per il 2022 (primo sottografico)
5 df_t['2022'].plot.box(ax=ax[0])
6
7 # Box plot per il 2021 (secondo sottografico)
8 df_t['2021'].plot.box(ax=ax[1])
9
10 # Calcola il minimo e il massimo dei valori del DataFrame
11 ymin = np.min(df_t.values)
12 ymax = np.max(df_t.values)
13
14 # Imposta i limiti dell'asse y (dal minimo al massimo) per entrambi i sottografici
15 for a in ax:
16     a.set_ylim(ymin-1, ymax+1)
17
18 # Mostra la figura
19 plt.show()

```



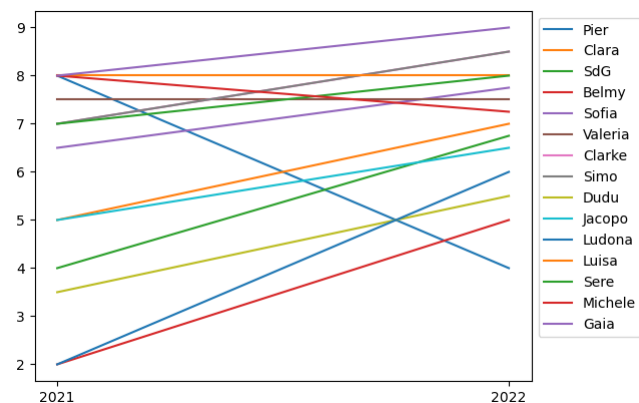
```
In [21]: 1 # Crea una figura con 15 sottografici
2 fig, ax = plt.subplots(15, 1, sharex=True, figsize=(8, 24))
3
4 # Crea il grafico a linee per ogni persona
5 for i, col in enumerate(df.columns):
6     ax[i].plot(df.index, df[col])
7     ax[i].set_ylabel(col)
8
9 # Mostra la figura
10 plt.show()
```




```

In [22]: 1 # Crea il grafico a linee per ogni persona
2 for col in df.columns:
3     plt.plot(df[col], label=col)
4
5 # Aggiungi le etichette dell'asse x
6 plt.xticks(np.arange(len(df)), df.index)
7
8 # Aggiungi la legenda in una posizione specifica
9 plt.legend(loc="upper left", bbox_to_anchor=(1,1))
10
11 # Mostra la figura
12 plt.show()

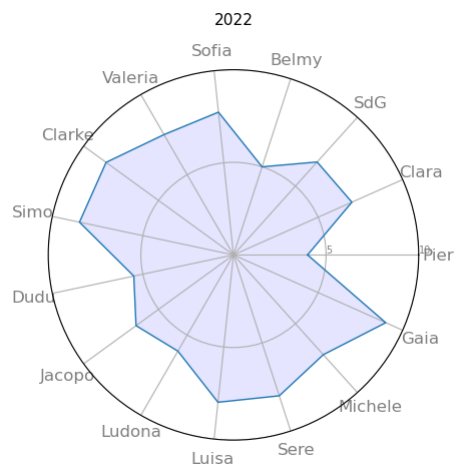
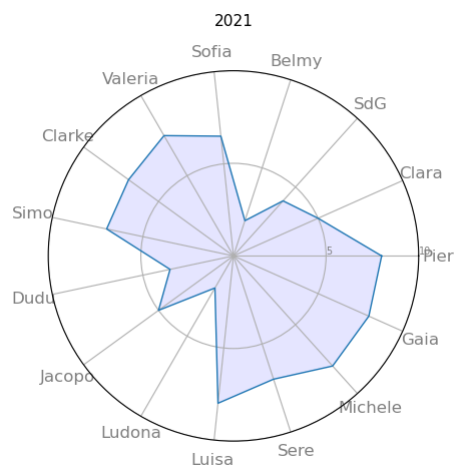
```



```

In [23]: 1 # Funzione per creare il grafico a radar
2 def radar(stats, title):
3     # Set dei valori
4     values = stats.values
5
6     # Numero di variabili
7     categories = list(stats.index)
8     N = len(categories)
9
10    # Aggiungi un valore fittizio alla fine del set di valori
11    values = np.append(values, values[0])
12
13    # Angoli di divisione delle sezioni
14    angles = [n / float(N) * 2 * pi for n in range(N)]
15    angles += angles[:1]
16
17    # Inizializza il grafico
18    ax = plt.subplot(111, polar=True)
19
20    # Disegna le linee di riferimento
21    plt.xticks(angles[:-1], categories, color='grey', size=12)
22    ax.set_rlabel_position(0)
23    plt.yticks([5, 10, 15], ["5", "10", "15"], color="grey", size=7)
24    plt.ylim(0, 15)
25
26    # Disegna le barre
27    ax.plot(angles, values, linewidth=1, linestyle='solid')
28    ax.fill(angles, values, 'b', alpha=0.1)
29    ax.set_rmax(10)
30
31    # Imposta il titolo
32    plt.title(title, size=11, y=1.1)
33
34    # Crea il grafico a radar per ogni persona
35    for col in df_t.columns:
36        radar(df_t[col], col)
37    plt.show()

```



```

In [24]: 1 # Crea il grafico a radar per ogni persona
2 fig, axs = plt.subplots(1, 2, figsize=(10, 5), sharey=True)
3 titles = df_t.columns
4
5 for ax, col, title in zip(axs, df_t.columns, titles):
6     radar(df_t[col], title)
7     ax.title.set_text(title)
8
9 # Aggiungi La Legenda
10 plt.legend(titles, loc="upper left", bbox_to_anchor=(1,1))
11
12 plt.show()

```

