- Shin's Lab -

Python for Data Visualization

Python for Data Visualization

-Chapter.4 Bar Plot -

4-00. Intro to Bar Plot

4-01. Bar Plot Basics

4-02. Multiple Bar Plots

4-03. Rect Objects

4-04. Horizontal Bar Plots

4-05. Exercises

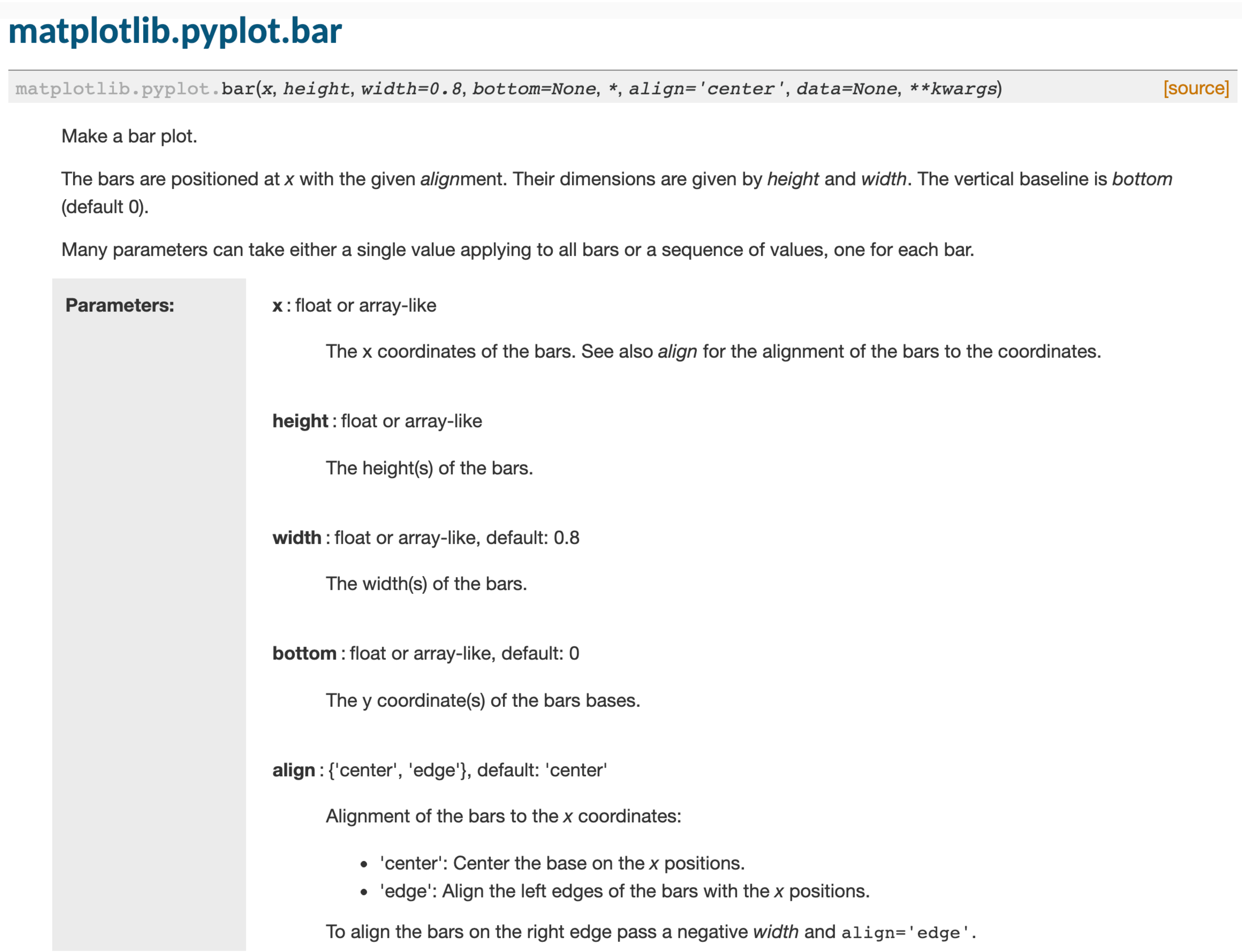
Python for Data Visualization

-Chapter.4 Bar Plot -

4-01. Bar Plot Basics

1. ax.bar Basics
2. Color and Hatch Customizing
3. Tick and Ticklabels
4. Bar Plot with Sorting
5. Bar Plot with OCED Data

1. ax.bar Basics



matplotlib.pyplot.bar

```
matplotlib.pyplot.bar(x, height, width=0.8, bottom=None, *, align='center', data=None, **kwargs)
```

[source]

Make a bar plot.

The bars are positioned at *x* with the given *align*ment. Their dimensions are given by *height* and *width*. The vertical baseline is *bottom* (default 0).

Many parameters can take either a single value applying to all bars or a sequence of values, one for each bar.

Parameters:

x : float or array-like

The x coordinates of the bars. See also *align* for the alignment of the bars to the coordinates.

height : float or array-like

The height(s) of the bars.

width : float or array-like, default: 0.8

The width(s) of the bars.

bottom : float or array-like, default: 0

The y coordinate(s) of the bars bases.

align : {'center', 'edge'}, default: 'center'

Alignment of the bars to the *x* coordinates:

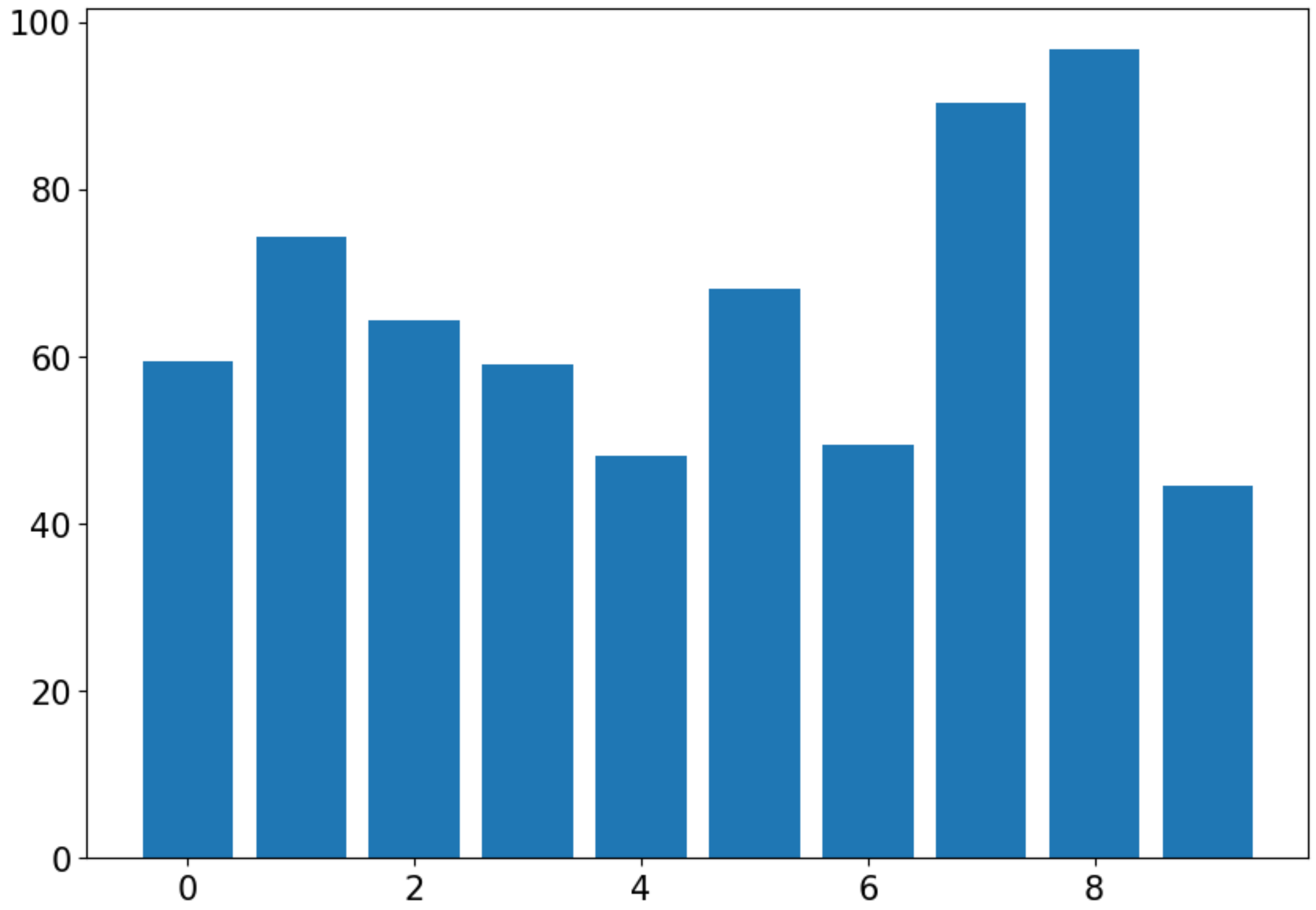
- 'center': Center the base on the *x* positions.
- 'edge': Align the left edges of the bars with the *x* positions.

To align the bars on the right edge pass a negative *width* and `align='edge'`.

https://matplotlib.org/3.3.1/api/_as_gen/matplotlib.pyplot.bar.html

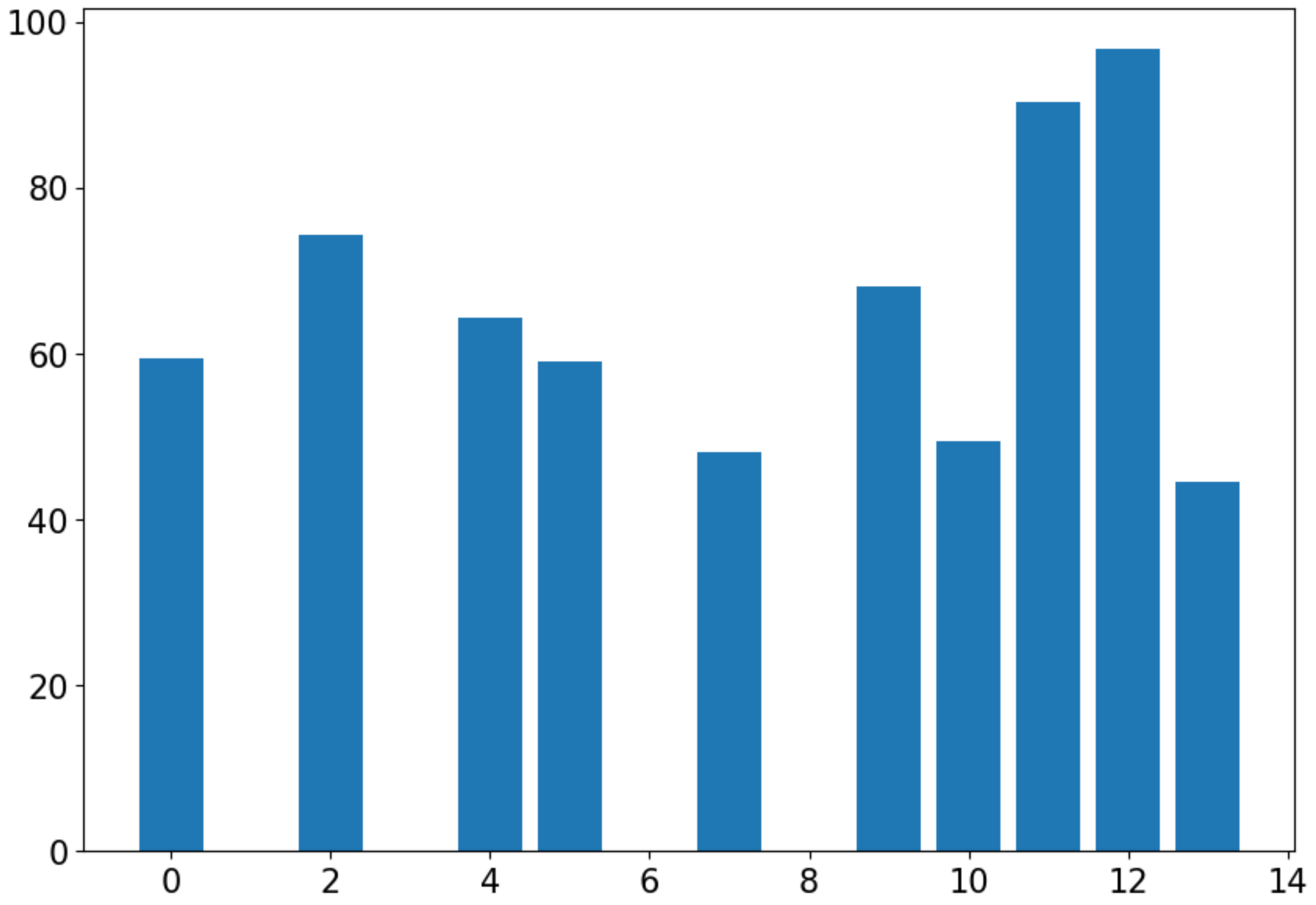
1. ax.bar Basics

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

n_data = 10
data = np.random.uniform(10, 100, (n_data,))
data_idx = np.arange(n_data)

fig, ax = plt.subplots(figsize=(10, 7))
ax.tick_params(labelsize=15)

ax.bar(data_idx, data)
```



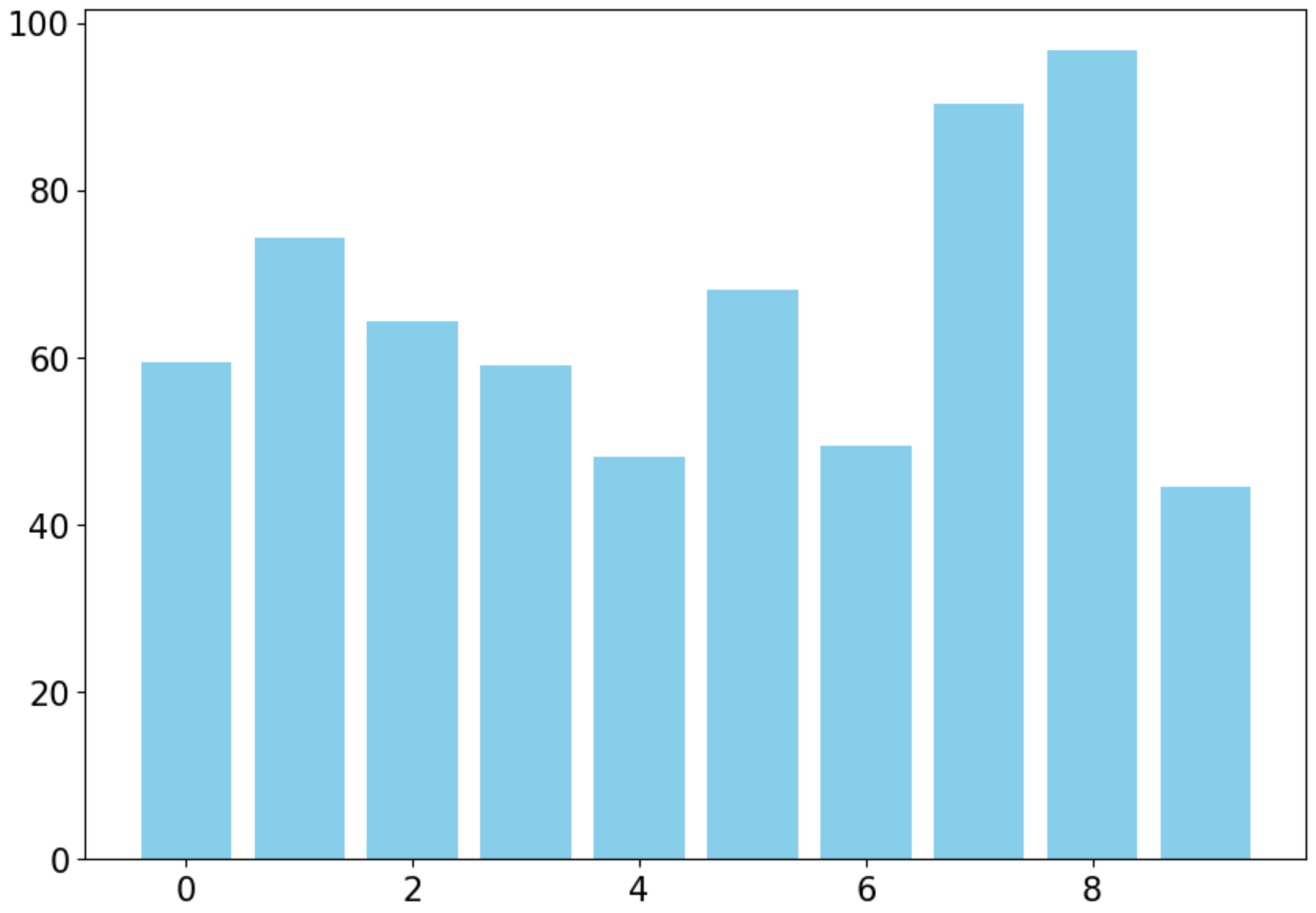
1. ax.bar Basics

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

data = np.random.uniform(10, 100, (10,))
data_idx = [0, 2, 4, 5, 7, 9, 10, 11, 12, 13]

fig, ax = plt.subplots(figsize=(10, 7))
ax.tick_params(labelsize=15)

ax.bar(data_idx, data)
```

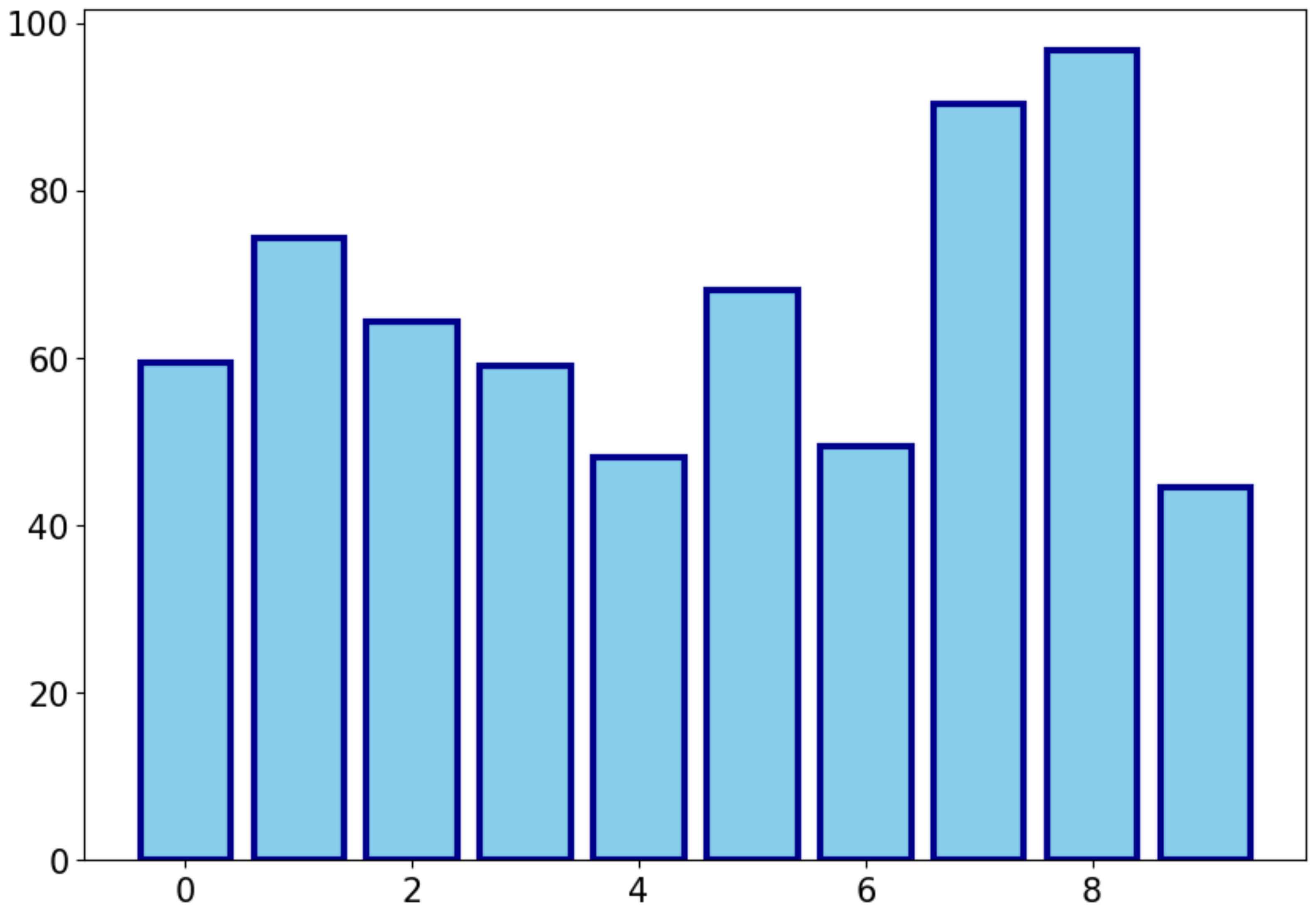


2. Color and Hatch Customizing

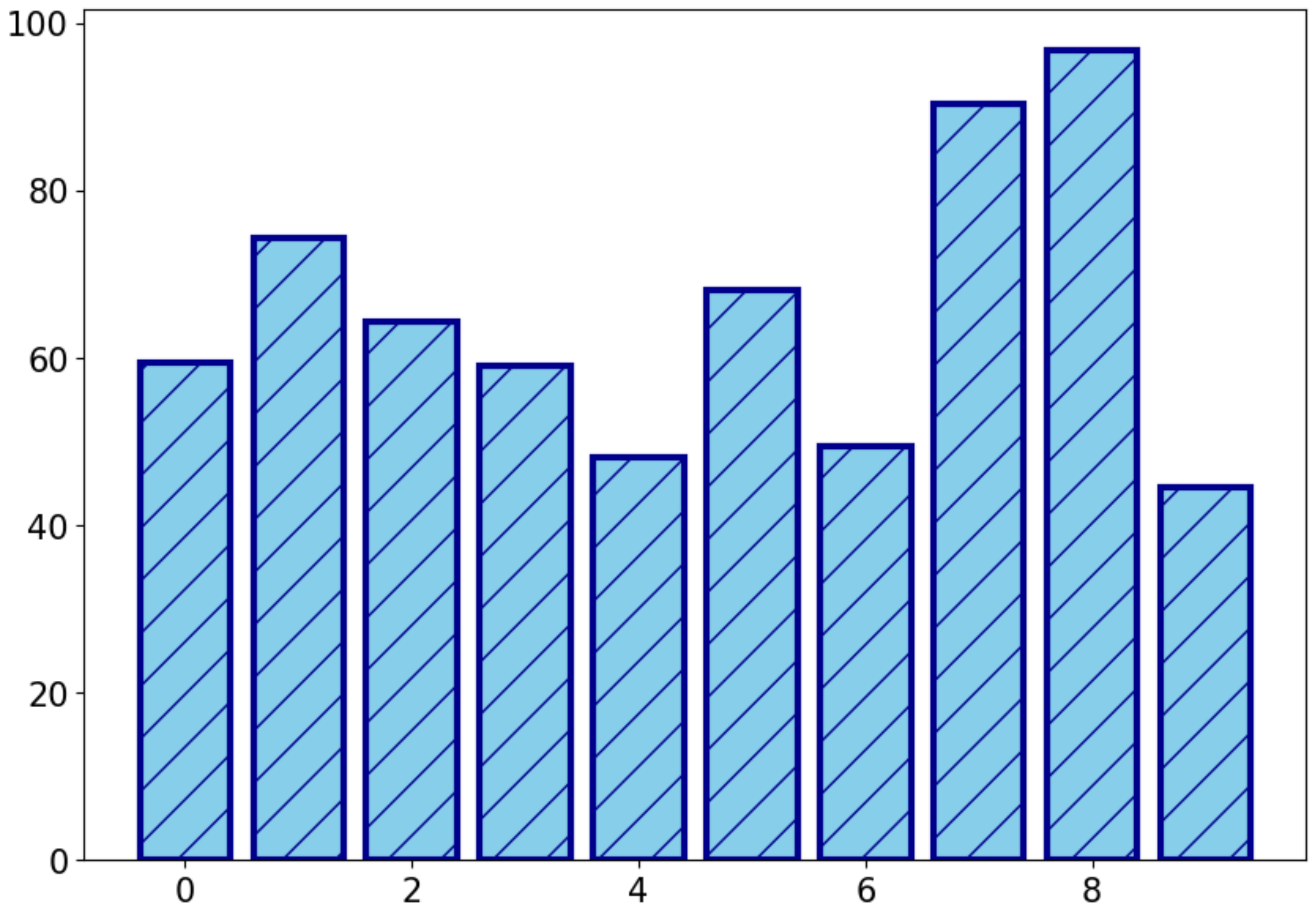
```
ax.bar(data_idx, data,
          facecolor='skyblue')
```



```
ax.bar(data_idx, data,
           facecolor='skyblue',
           edgecolor='darkblue',
           linewidth=3)
```

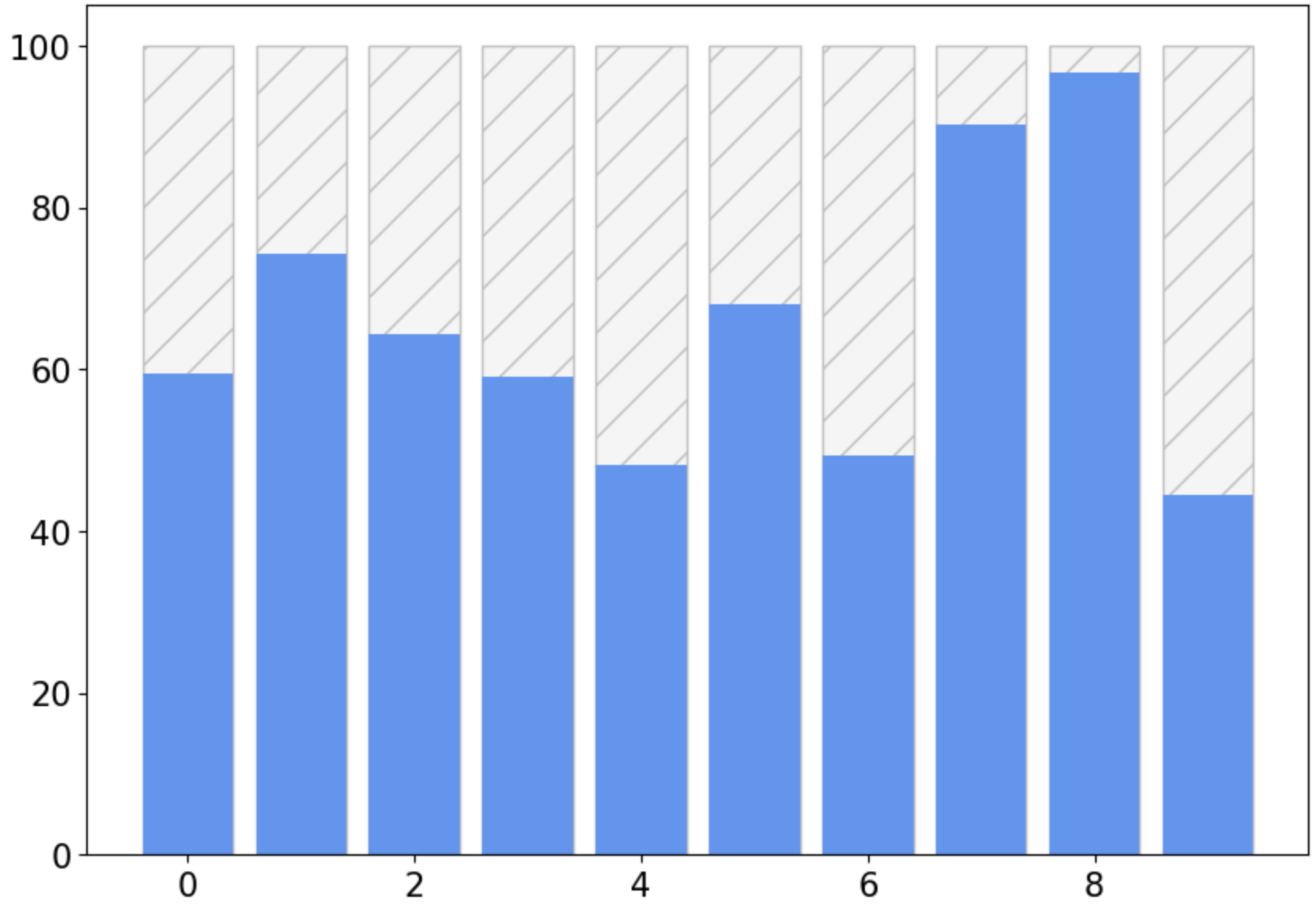


```
ax.bar(data_idx, data,
           facecolor='skyblue',
           edgecolor='darkblue',
           linewidth=3,
           hatch='/')
```



2. Color and Hatch Customizing

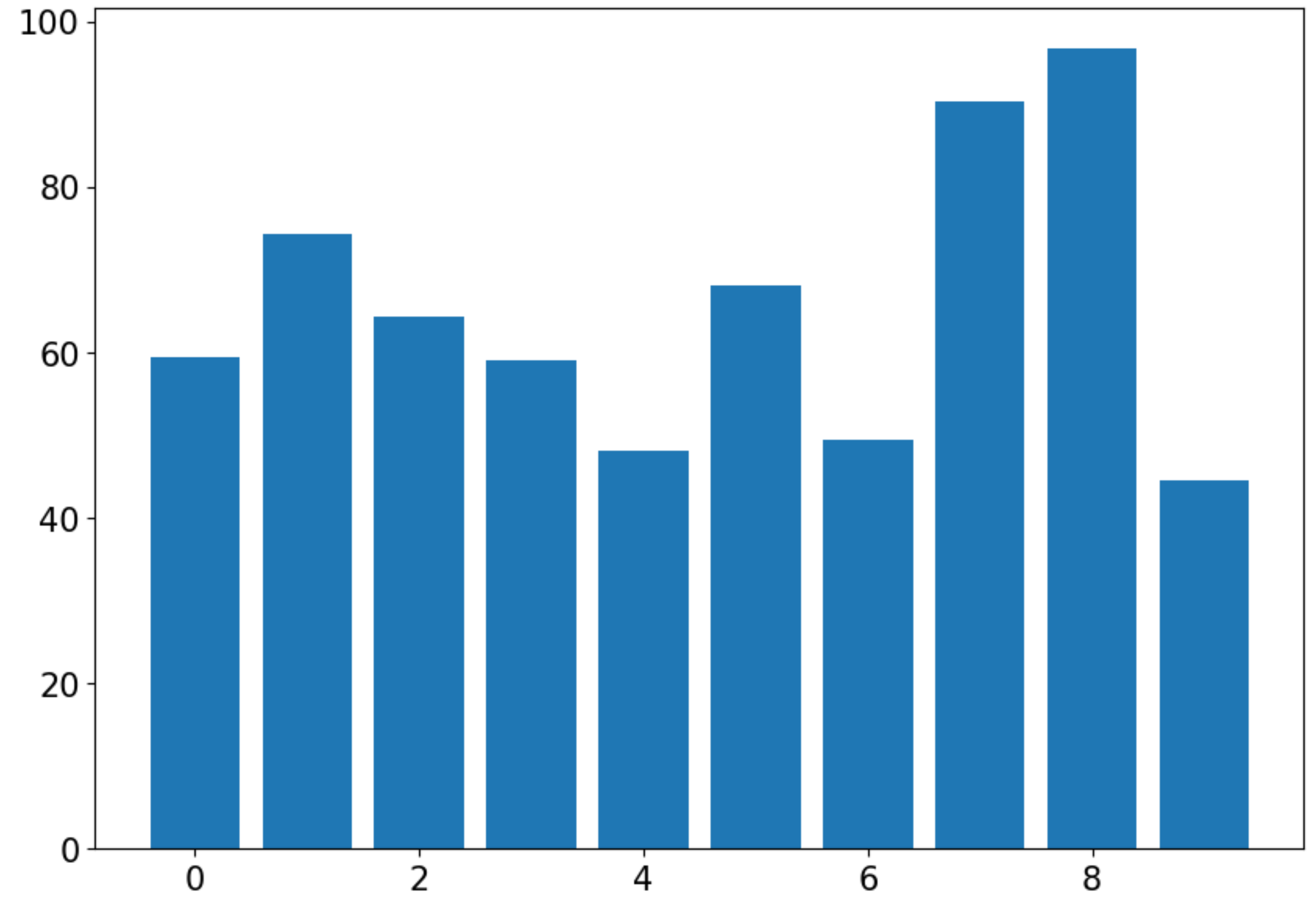
```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

n_data = 10
data = np.random.uniform(10, 100, (n_data,))
background = 100*np.ones(n_data)
data_idx = np.arange(n_data)

fig, ax = plt.subplots(figsize=(10, 7))
ax.tick_params(labelsize=15)

ax.bar(data_idx, background,
       facecolor='whitesmoke',
       hatch='/',
       edgecolor='silver')

ax.bar(data_idx, data,
       facecolor='cornflowerblue')
```

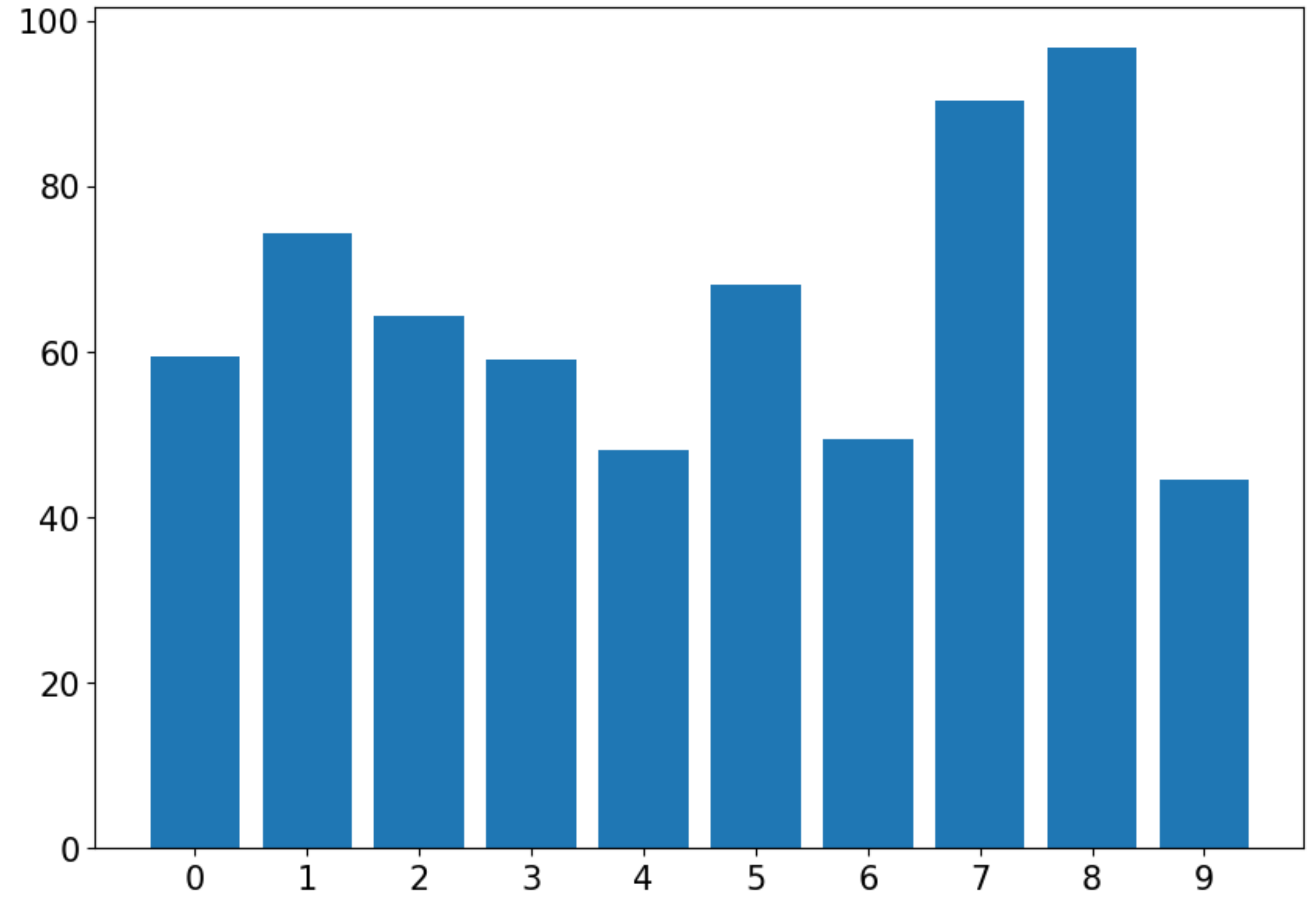


3. Tick and Ticklabels

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

n_data = 10
data = np.random.uniform(10, 100, (n_data,))
data_idx = np.arange(n_data)

fig, ax = plt.subplots(figsize=(10, 7))
ax.tick_params(labelsize=15)

ax.bar(data_idx, data)
```

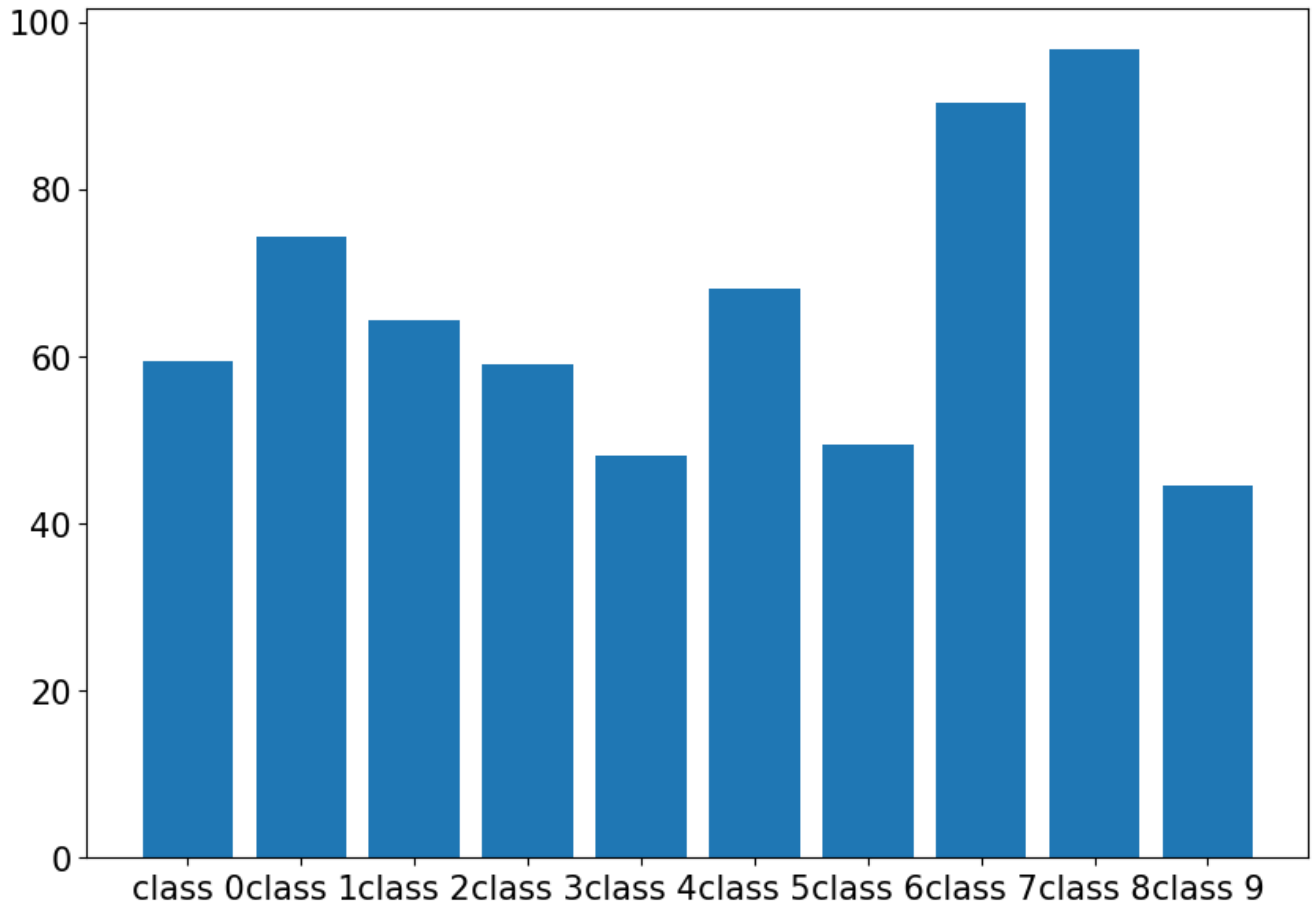
```
ax.bar(data_idx, data,
       tick_label=data_idx)
```





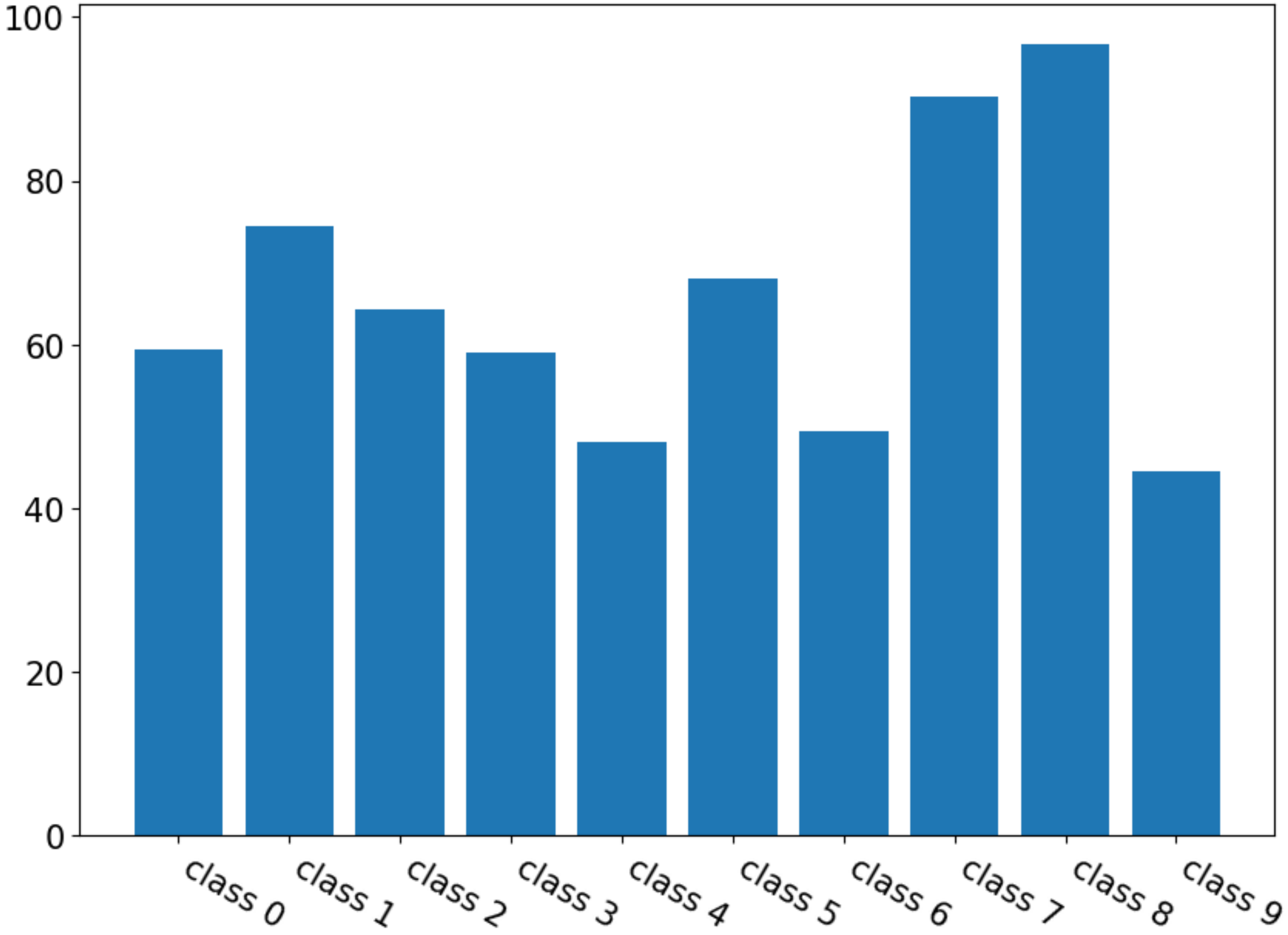
3. Tick and Ticklabels

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

n_data = 10
data = np.random.uniform(10, 100, (n_data,))
data_idx = np.arange(n_data)

fig, ax = plt.subplots(figsize=(10, 7))
ax.tick_params(labelsize=15)


data_labels = ['class ' + str(i) for i in range(n_data)]
ax.bar(data_idx, data,
       tick_label=data_labels)
```



3. Tick and Ticklabels

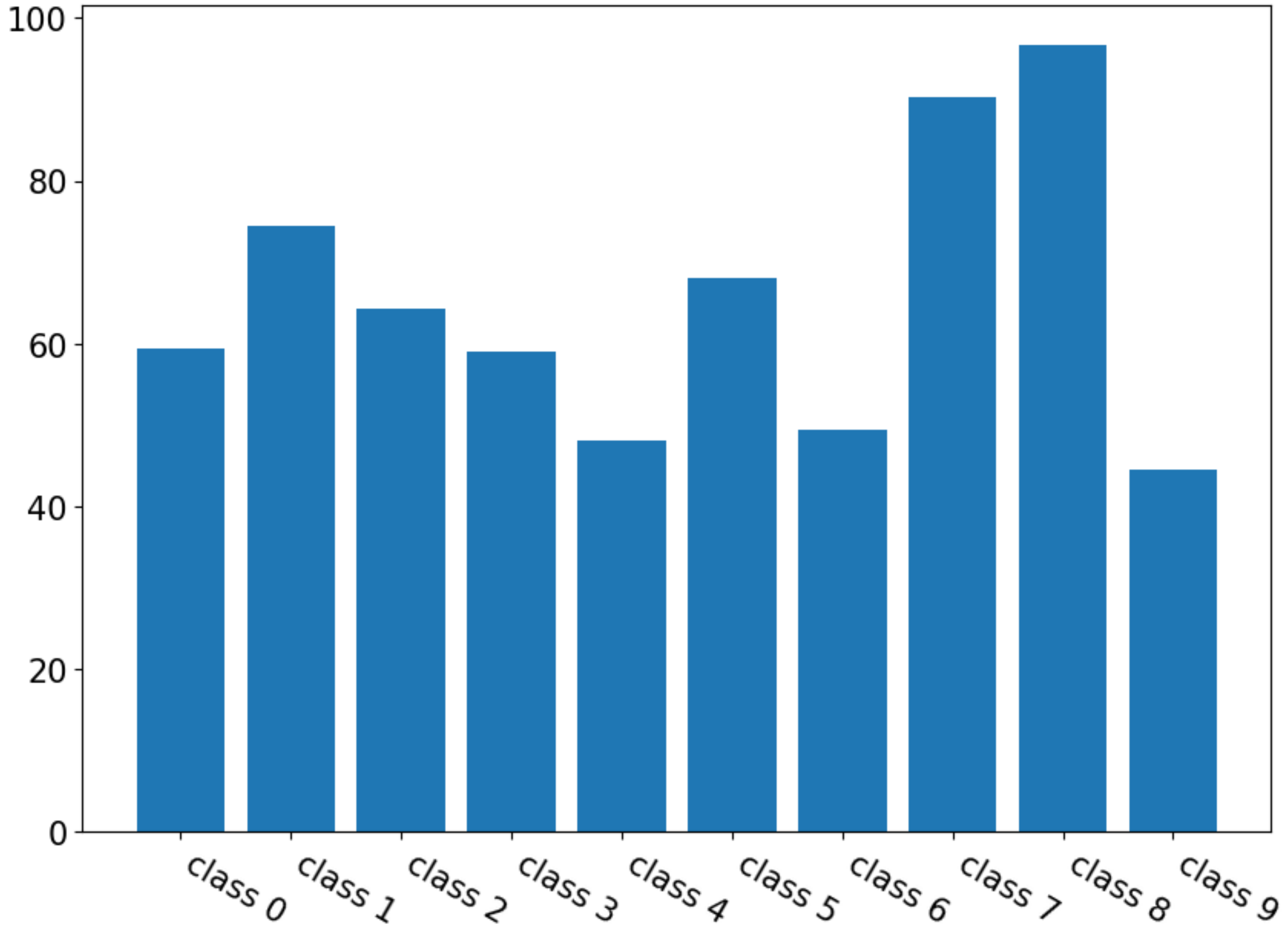
```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

n_data = 10
data = np.random.uniform(10, 100, (n_data,))
data_idx = np.arange(n_data)

fig, ax = plt.subplots(figsize=(10, 7))
ax.tick_params(labelsize=15)


data_labels = ['class ' + str(i) for i in range(n_data)]
ax.bar(data_idx, data)
ax.set_xticks(data_idx)
ax.set_xticklabels(data_labels,
                   rotation=-30,
                   ha='left')
```

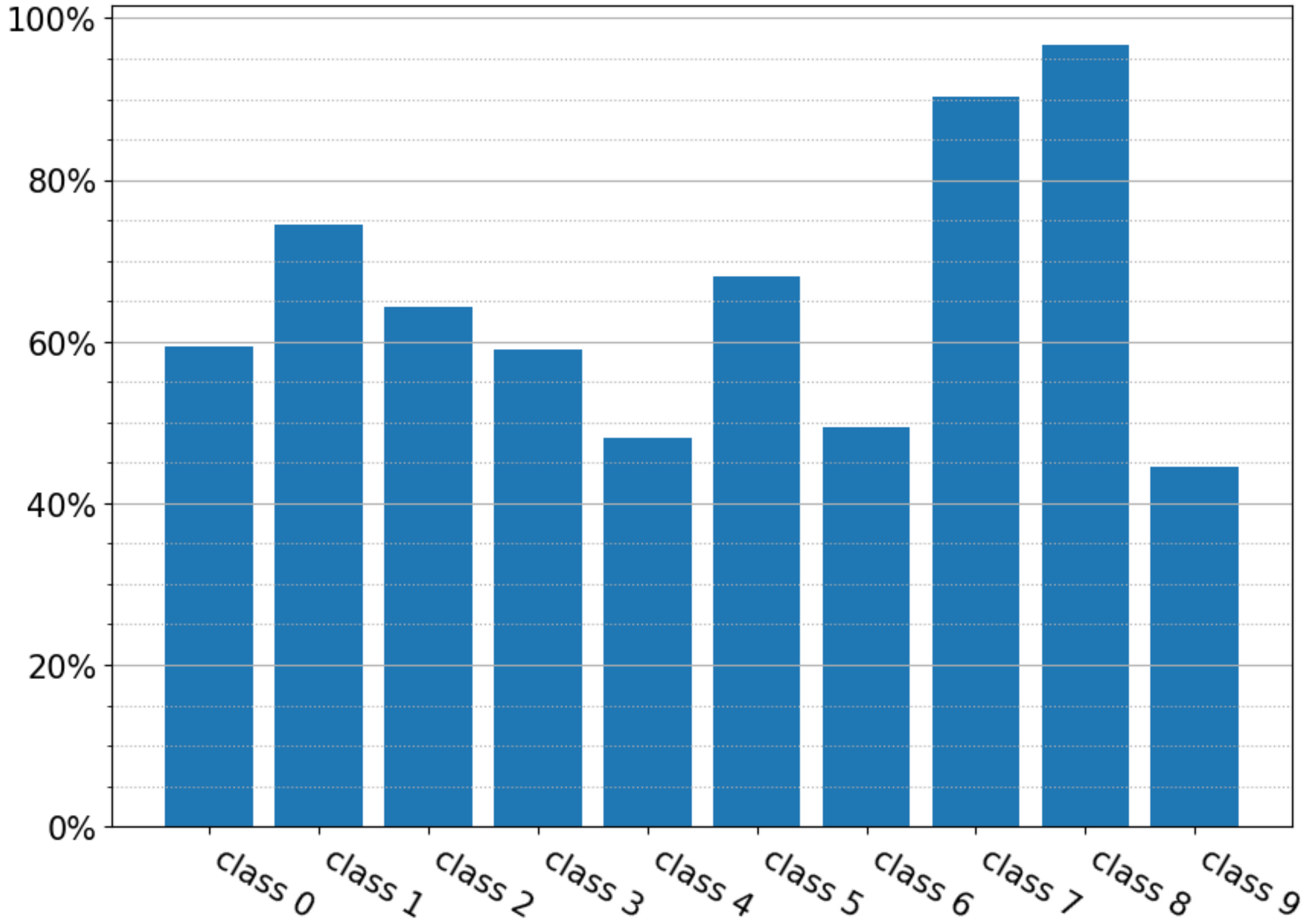


3. Tick and Ticklabels

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

n_data = 10
data = np.random.uniform(10, 100, (n_data,))
data_idx = np.arange(n_data)

fig, ax = plt.subplots(figsize=(10, 7))
ax.tick_params(labelsize=15)

data_labels = ['class ' + str(i) for i in range(n_data)]
ax.bar(data_idx, data)
ax.set_xticks(data_idx)
ax.set_xticklabels(data_labels,
                   rotation=-30,
                   ha='left')
```

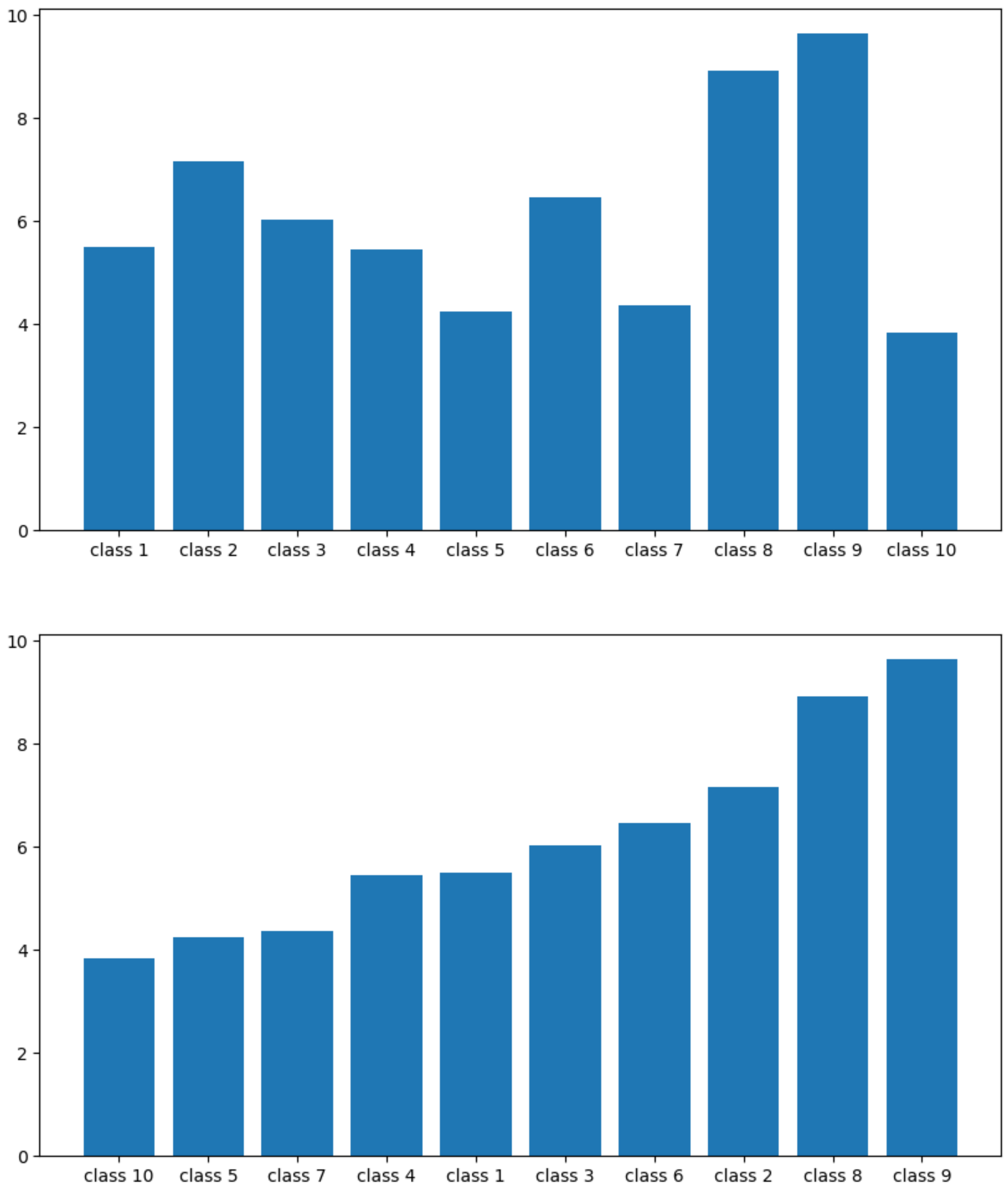
```
major_yticks = np.arange(0, 101, 20)
major_yticklabels = [str(p) + '%' for p in major_yticks]
minor_yticks = np.arange(0, 101, 5)

ax.set_yticks(major_yticks)
ax.set_yticklabels(major_yticklabels)
ax.set_yticks(minor_yticks,
              minor=True)

ax.grid(axis='y',
        which='major')
ax.grid(axis='y',
        which='minor',
        linestyle=':')
```





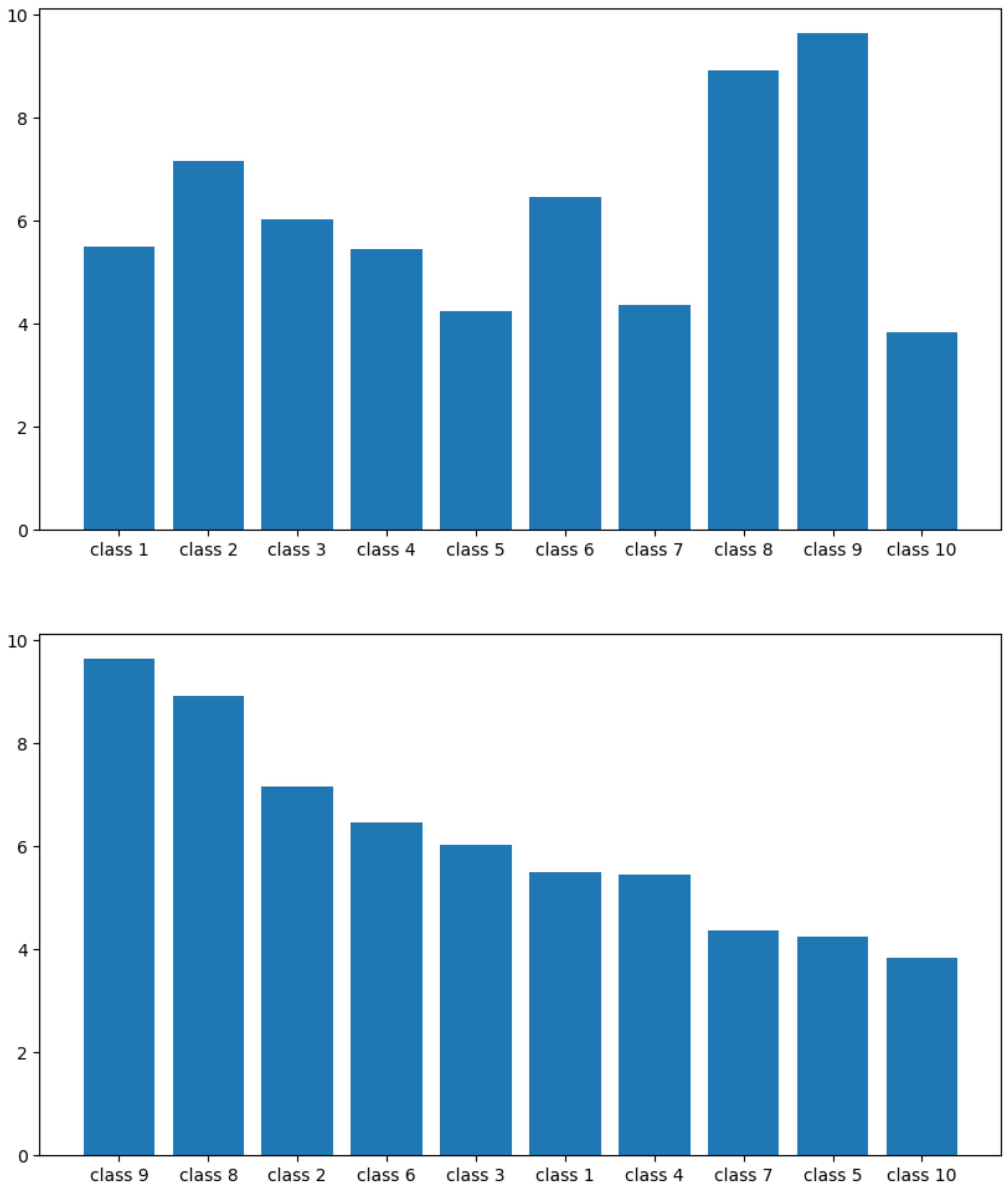
4. Bar Plot with Sorting

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

n_data = 10
data = np.random.uniform(0, 10, n_data)
data_idx = np.arange(n_data)
data_labels = np.array(['class ' + str(i+1) for i in range(n_data)])

sort_idx = np.argsort(data)
data_sort = data[sort_idx]
data_labels_sort = data_labels[sort_idx]

fig, axes = plt.subplots(2, 1, figsize=(10, 12))
axes[0].bar(data_idx, data,
            tick_label=data_labels)

axes[1].bar(data_idx, data_sort, # not "sort_idx"
            tick_label=data_labels_sort)
```



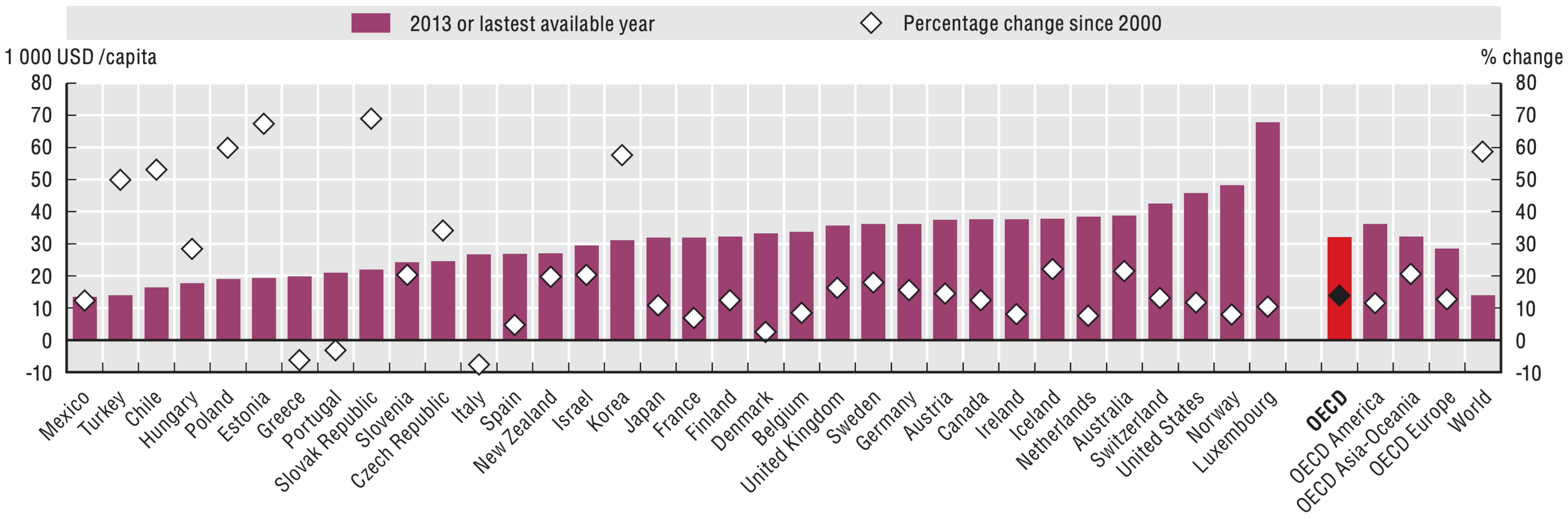
4. Bar Plot with Sorting

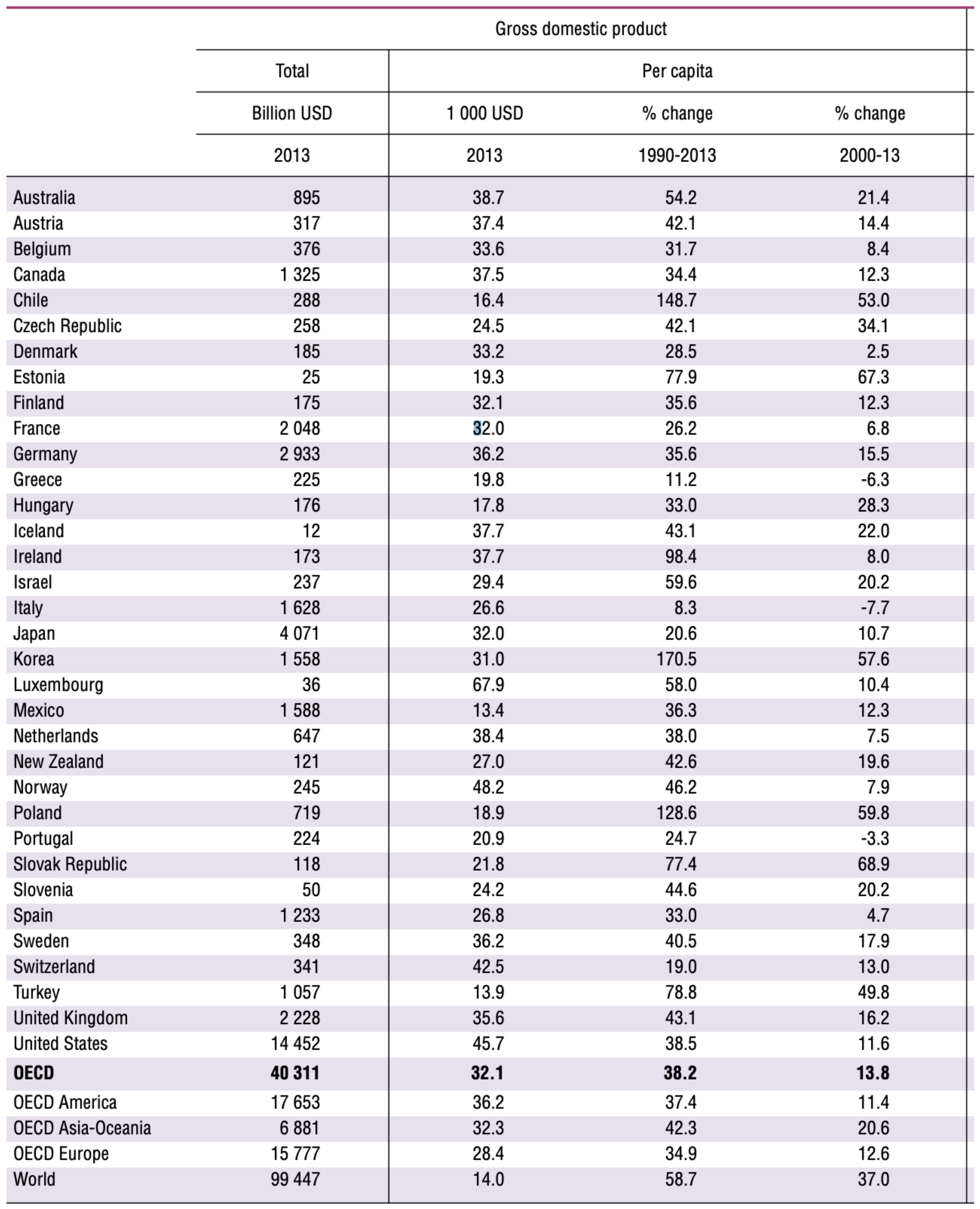
```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

n_data = 10
data = np.random.uniform(0, 10, n_data)
data_idx = np.arange(n_data)
data_labels = np.array(['class ' + str(i+1) for i in range(n_data)])



sort_idx = np.argsort(data)[::-1]
data_sort = data[sort_idx]
data_labels_sort = data_labels[sort_idx]


fig, axes = plt.subplots(2, 1, figsize=(10, 12))
axes[0].bar(data_idx, data,
            tick_label=data_labels)


axes[1].bar(data_idx, data_sort, # not "sort_idx"
            tick_label=data_labels_sort)
```



5. Bar Plot with OCED Data

Figure 2.20. **Gross Domestic Product (GDP) per capita, 2013**





	Gross domestic product			
	Total	Per capita		
	Billion USD	1 000 USD	% change	% change
	2013	2013	1990-2013	2000-13
Australia	895	38.7	54.2	21.4
Austria	317	37.4	42.1	14.4
Belgium	376	33.6	31.7	8.4
Canada	1 325	37.5	34.4	12.3
Chile	288	16.4	148.7	53.0
Czech Republic	258	24.5	42.1	34.1
Denmark	185	33.2	28.5	2.5
Estonia	25	19.3	77.9	67.3
Finland	175	32.1	35.6	12.3
France	2 048	32.0	26.2	6.8
Germany	2 933	36.2	35.6	15.5
Greece	225	19.8	11.2	-6.3
Hungary	176	17.8	33.0	28.3
Iceland	12	37.7	43.1	22.0
Ireland	173	37.7	98.4	8.0
Israel	237	29.4	59.6	20.2
Italy	1 628	26.6	8.3	-7.7
Japan	4 071	32.0	20.6	10.7
Korea	1 558	31.0	170.5	57.6
Luxembourg	36	67.9	58.0	10.4
Mexico	1 588	13.4	36.3	12.3
Netherlands	647	38.4	38.0	7.5
New Zealand	121	27.0	42.6	19.6
Norway	245	48.2	46.2	7.9
Poland	719	18.9	128.6	59.8
Portugal	224	20.9	24.7	-3.3
Slovak Republic	118	21.8	77.4	68.9
Slovenia	50	24.2	44.6	20.2
Spain	1 233	26.8	33.0	4.7
Sweden	348	36.2	40.5	17.9
Switzerland	341	42.5	19.0	13.0
Turkey	1 057	13.9	78.8	49.8
United Kingdom	2 228	35.6	43.1	16.2
United States	14 452	45.7	38.5	11.6
OECD	**40 311**	**32.1**	**38.2**	**13.8**
OECD America	17 653	36.2	37.4	11.4
OECD Asia-Oceania	6 881	32.3	42.3	20.6
OECD Europe	15 777	28.4	34.9	12.6
World	99 447	14.0	58.7	37.0

Source: OECD (2015), *OECD Historical Population Data and Projections* (database); OECD (2015), OECD National Account

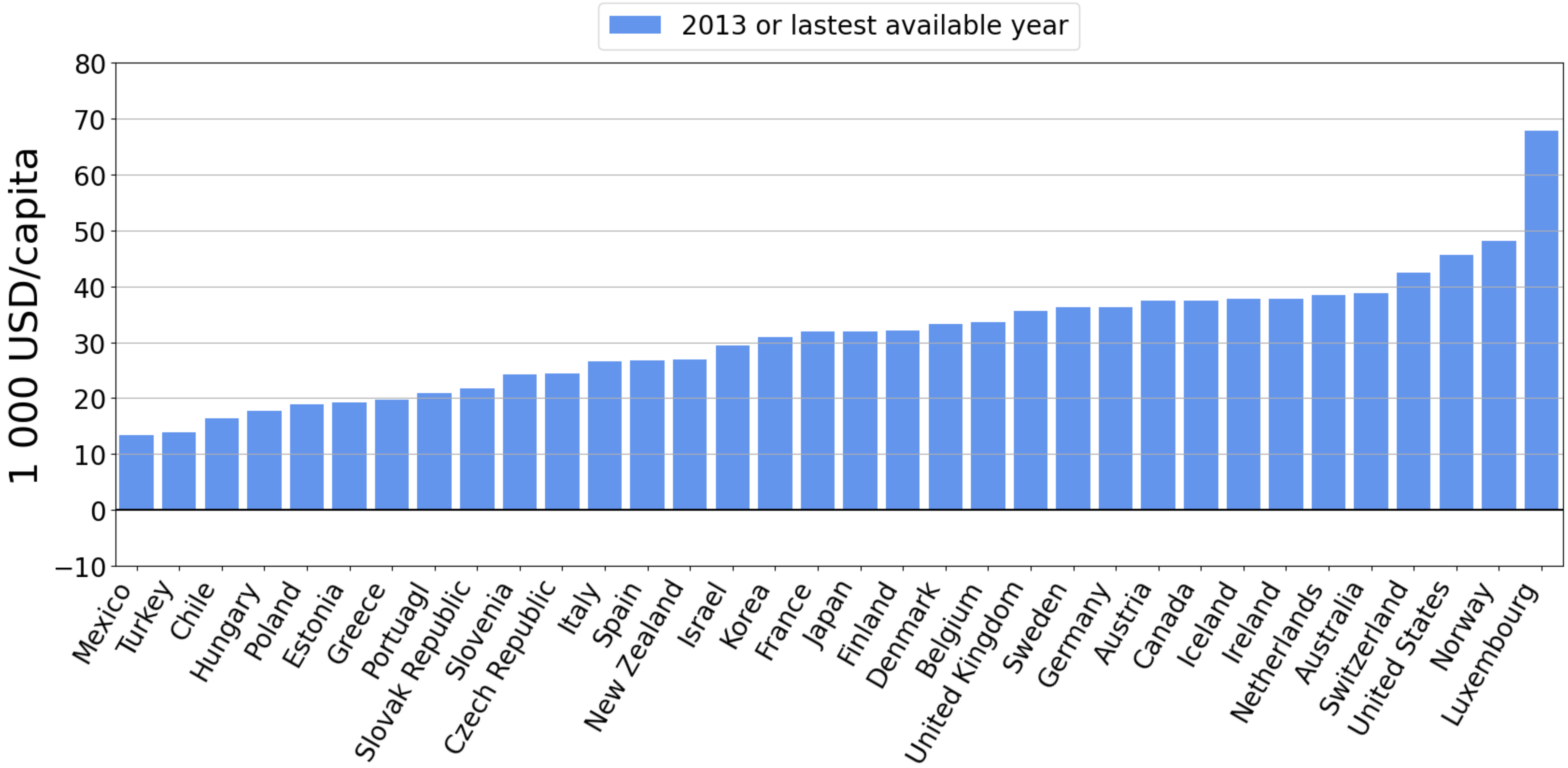
ENVIRONMENT AT A GLANCE 2015: OECD INDICATORS © OECD 2015

5. Bar Plot with OCED Data

```python
import matplotlib.pyplot as plt
import numpy as np

countries = np.array(['Australia', 'Austria', 'Belgium', 'Canada', 'Chile',
                      'Czech Republic', 'Denmark', 'Estonia', 'Finland', 'France',
                      'Germany', 'Greece', 'Hungary', 'Iceland', 'Ireland',
                      'Israel', 'Italy', 'Japan', 'Korea', 'Luxembourg',
                      'Mexico', 'Netherlands', 'New Zealand', 'Norway', 'Poland',
                      'Portuagl', 'Slovak Republic', 'Slovenia', 'Spain', 'Sweden',
                      'Switzerland', 'Turkey', 'United Kingdom', 'United States'])

gdp = np.array([38.7, 37.4, 33.6, 37.5, 16.4,
                24.5, 33.2, 19.3, 32.1, 32.0,
                36.2, 19.8, 17.8, 37.7, 37.7,
                29.4, 26.6, 32.0, 31.0, 67.9,
                13.4, 38.4, 27.0, 48.2, 18.9,
                20.9, 21.8, 24.2, 26.8, 36.2,
                42.5, 13.9, 35.6, 45.7])

gdp_p = np.array([21.4, 14.4, 8.4, 12.3, 53.0,
                  34.1, 2.5, 67.3, 12.3, 6.8,
                  15.5, -6.3, 28.3, 22.0, 8.0,
                  20.2, -7.7, 10.7, 57.6, 10.4,
                  12.3, 7.5, 19.6, 7.9, 59.8,
                  -3.3, 68.9, 20.2, 4.7, 17.9,
                  13.0, 49.8, 16.2, 11.6])

gdp_idx = np.arange(len(gdp))

idx_sort = np.argsort(gdp)
gdp_sort = gdp[idx_sort]
countries_sort = countries[idx_sort]
```

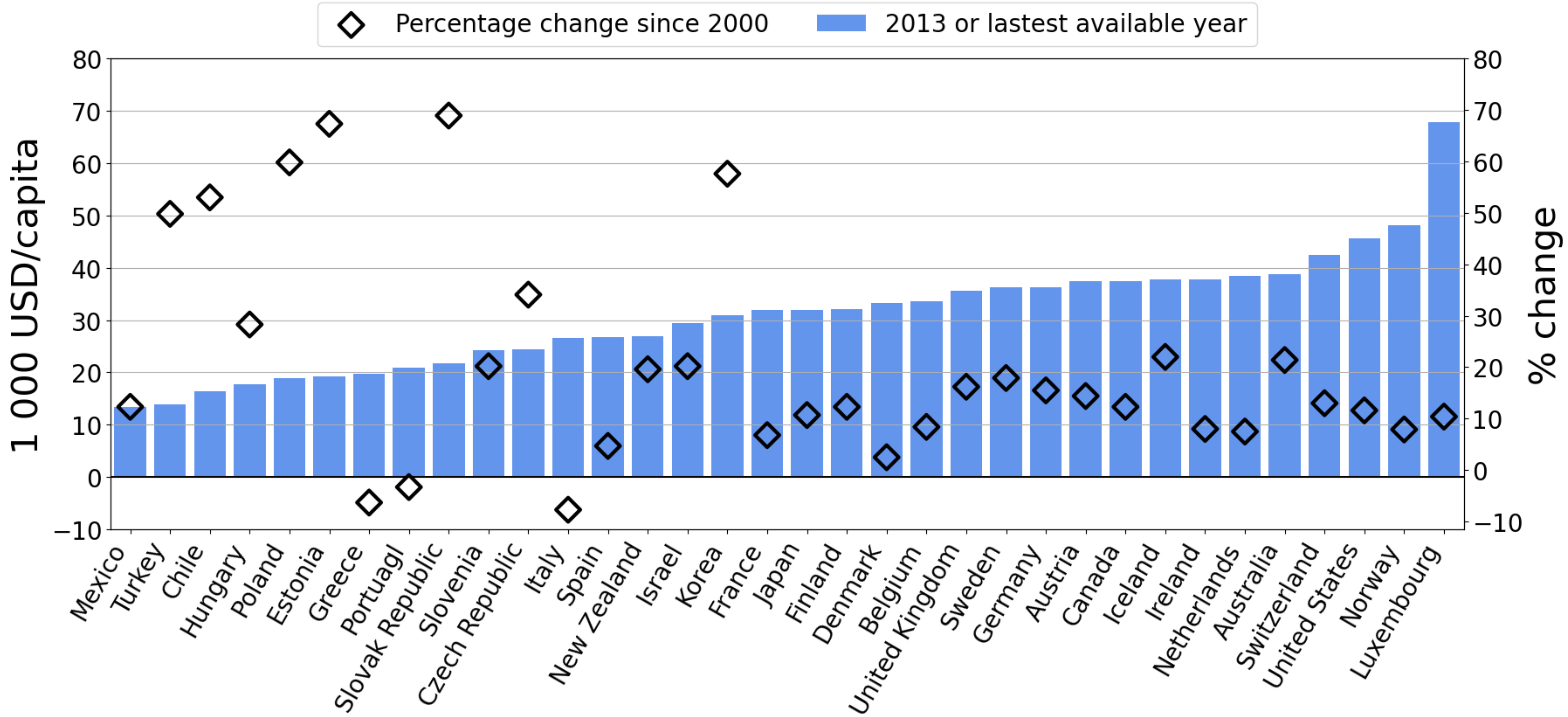
```python
fig, ax = plt.subplots(figsize=(20, 7))
ax.bar(gdp_idx, gdp_sort,
       color='cornflowerblue',
       label='2013 or lastest available year')
ax.legend(loc='lower center',
          bbox_to_anchor=(0.5, 1),
          fontsize=20)

yticks = np.arange(-10, 81, 10)
ax.set_xticks(gdp_idx)
ax.set_xticklabels(countries_sort,
                   rotation=60,
                   ha='right')
ax.set_yticks(yticks)

ax.tick_params(labelsize=20)

ax.set_xlim([-0.5, 33.5])
ax.set_ylim([-10, 80])
ax.axhline(y=0,
           color='black')
ax.set_ylabel('1 000 USD/capita',
              fontsize=30)
ax.grid(axis='y')
```

5. Bar Plot with OCED Data



5. Bar Plot with OCED Data

```
ax2 = ax.twinx()
ax2.scatter(gdp_idx, gdp_p_sort,
            marker='D',
            s=200,
            edgecolor='black',
            facecolor='None',
            linewidth=3)
ax2.set_yticks(yticks)
ax2.tick_params(labelsize=20)
ax2.set_ylabel('% change',
               fontsize=30)


ax.scatter([], [],
            marker='D',
            s=200,
            edgecolor='black',
            facecolor='None',
            linewidth=3,
            label='Percentage change since 2000')

ax.legend(loc='lower center',
          bbox_to_anchor=(0.5, 1),
          fontsize=20,
          ncol=2)
```



Python for Data Visualization

-Chapter.4 Bar Plot -

4-01. Bar Plot Basics

1. ax.bar Basics
2. Color and Hatch Customizing
3. Tick and Ticklabels
4. Bar Plot with Sorting
5. Bar Plot with OCED Data