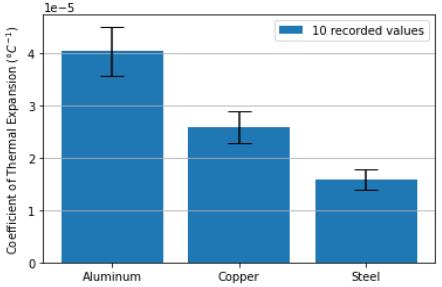
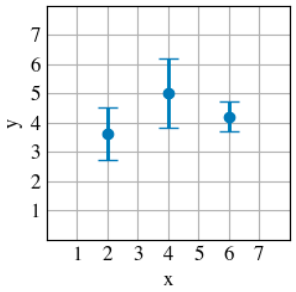
Visualizing errors

`ax.bar(x,mean_values, yerr=stderr)`	bar graph with error bar	compare data and errors between different groups	
`ax.`**`errorbar`**`(x,y,yerr)`	error bar	Visualize data and errors	

The error bar usually reports the standard error, where σ is the standard deviation, and N the number of samples

$$SE = \frac{\sigma}{\sqrt{N}}$$

Look at the documentation page of **errorbar()**
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.errorbar.html
and **bar()** to make a bar graph with error bars
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html

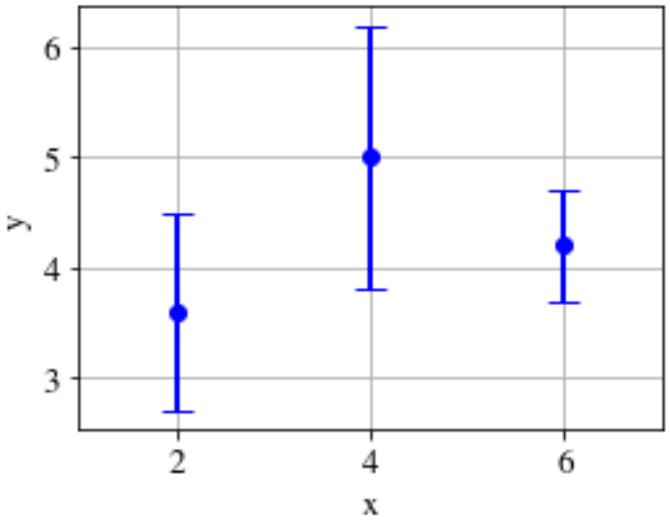
Example – errorbar()

```
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams['font.size']=13

#some fictitious data from a fictitious experiment.
x = [2, 4, 6]
y = [3.6, 5, 4.2]
yerr = [0.9, 1.2, 0.5]

# plot
fig, ax = plt.subplots(figsize=(4,3))
ax.errorbar(x, y, yerr, fmt='o', linewidth=2, capsize=6, color='b')
ax.set_xlim(1,7)
ax.ylim=(0, 8)
ax.xticks=np.arange(1, 8)
ax.yticks=np.arange(1, 8)

ax.set_xlabel('x')
ax.set_ylabel('y')
ax.grid()
plt.show()
```



Example – bar graph with error bars

We use the measured coefficient of thermal expansion (CTE) of three metals: Aluminum, Copper, and Steel. The unit for coefficient of thermal expansion is per degrees Celsius (/ °C). We calculate the mean and standard error for each metal and use a bar graph where we also visualize the standard error.

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

df=pd.read_csv('data-metals.csv') # find the data set on Canvas/DATA
print(df)

#calculate the average for each metal/column
mean_metal=df.mean(axis=0)

#calculate the standar error for each metal/column
std=df.std(axis=0)
std_error=std/np.sqrt(len(df))

#make the bar graph with error bars
fig, ax = plt.subplots(figsize=(5,4))
plt.rcParams['font.size']=13
names=df.columns.values
ax.bar(names,mean_metal, yerr=std_error, capsize=10, label='10 recorded
values')
ax.set_ylabel('Coefficient of Thermal Expansion ($\degree C^{-1}$)')
ax.grid(axis='y')
ax.legend()
```