

```
In [1]: %%html  
<link rel="stylesheet" type="text/css" href="rise.css" />
```

Data visualization

There are multiple options of plotting data in Python.

Matplotlib is one of the most widely used and allows you to customize nearly every aspect of your plots. A downside is that the default settings don't look the best.

There are more options, but I'll let you explore them on your own.

Install Matplotlib

In a cmd shell or terminal run:

```
conda activate neu365  
pip install matplotlib
```

Let's check out some Matplotlib example plot types

I expect that you will be able to use the [online documentation](#) and [google](#) to [figure out how to make whatever plots you want](#).

Here I provide you with a [quick start](#) tutorial to get you going.

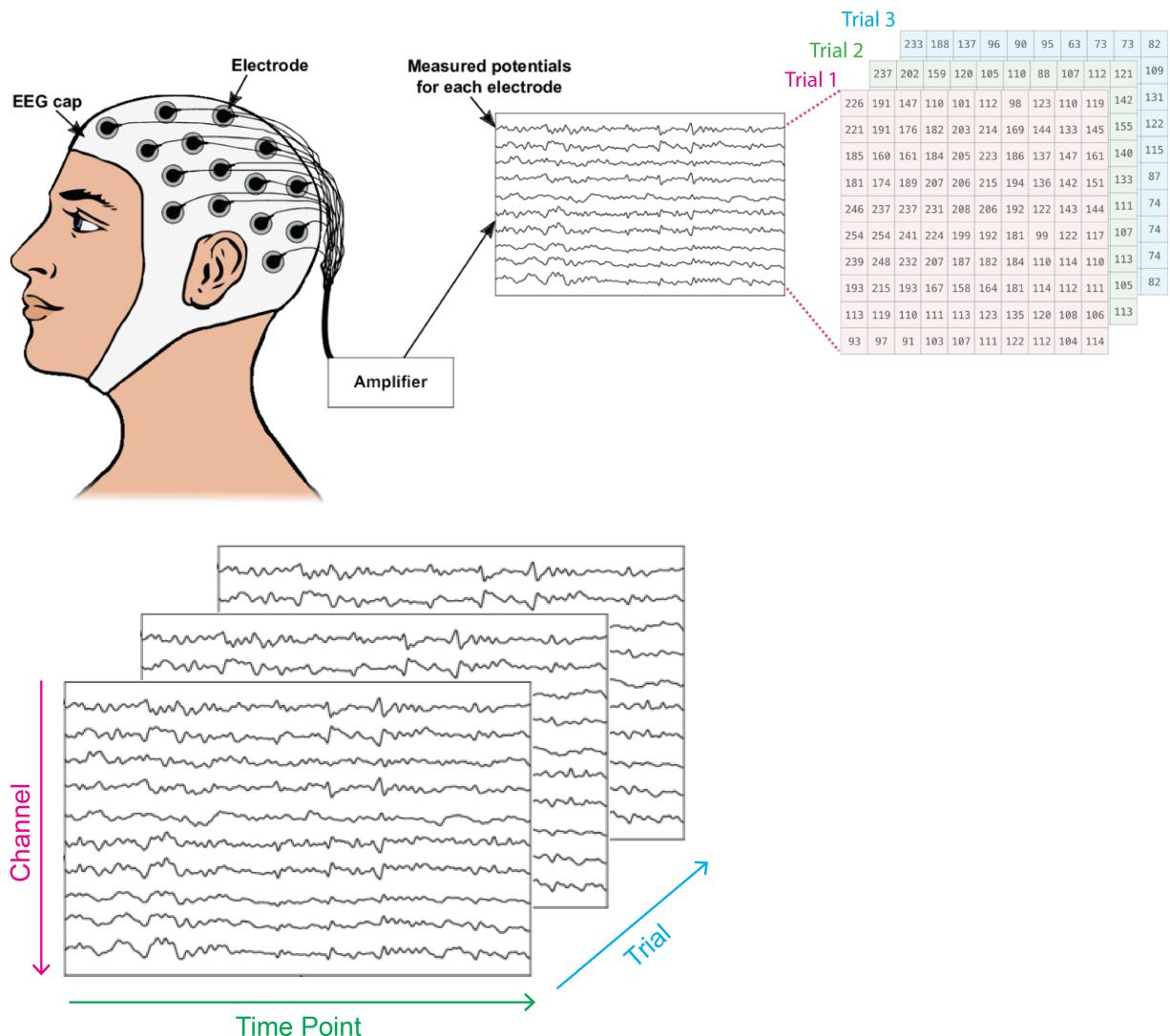
A Quick Start guide for basic plots in Matplotlib

Learning goals

- You will be able to plot 2-D data.
- You will be able to style your plots.
- You will be able to display grids of plot.

Let's consider a 3-D array of EEG *time series* for multiple

channels, trials



```
In [2]: # Load EEG data from file

import numpy as np

# See np.load & np.save
# Filepath is relative to this notebook
EEGs = np.load("data/EEGs.npy")

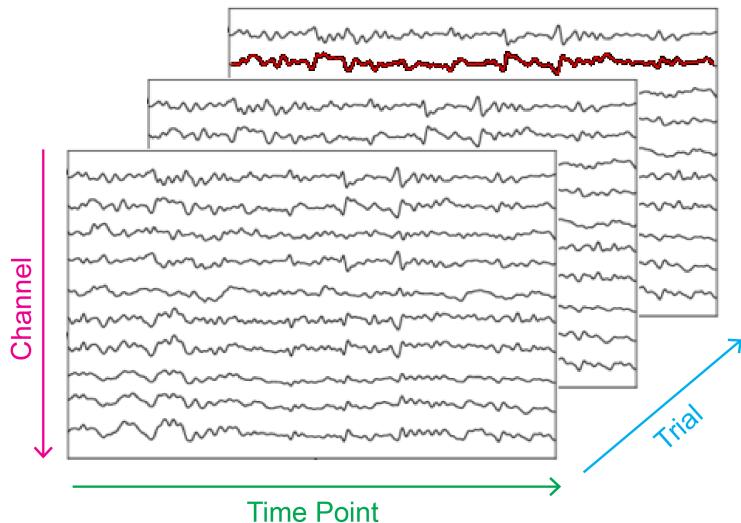
EEGs.shape
```

Out[2]: (64, 640, 99)

```
In [3]: print(f"# channels = {EEGs.shape[0]}")
print(f"# time pts = {EEGs.shape[1]}")
print(f"# trials    = {EEGs.shape[2]}")
```

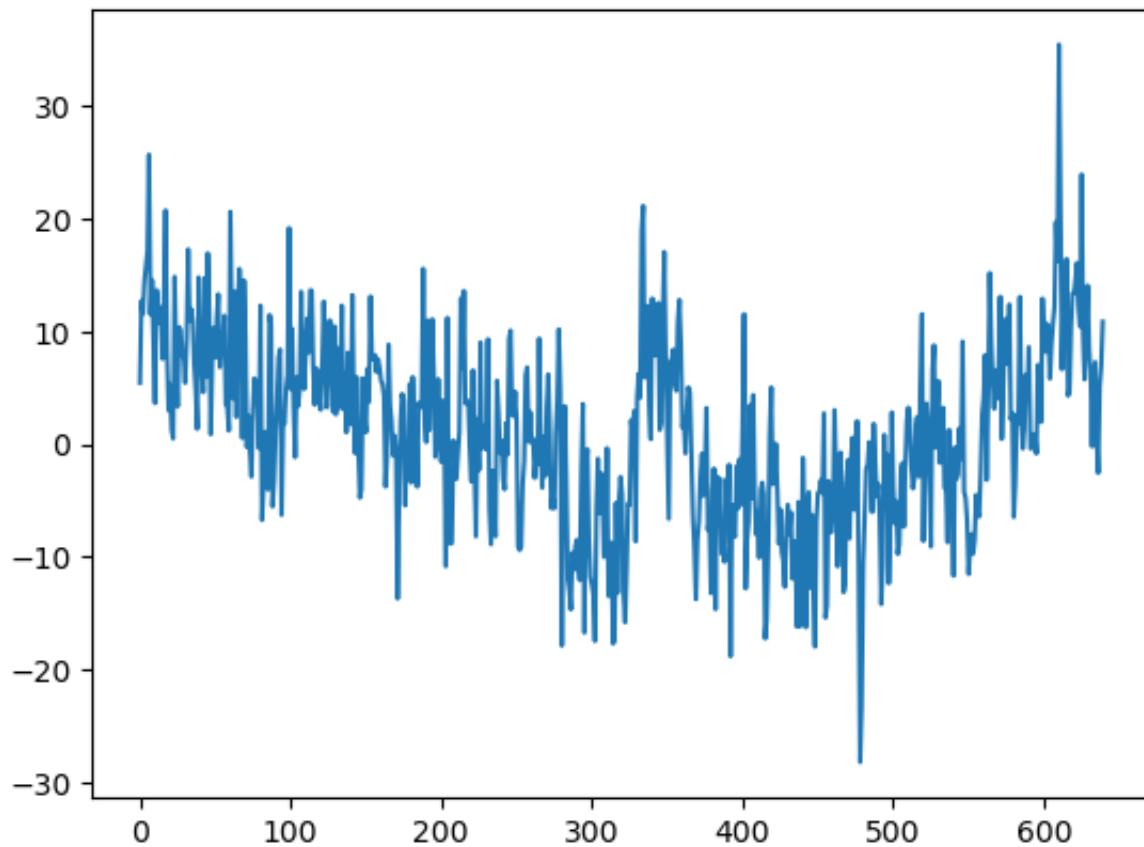
```
# channels = 64
# time pts = 640
# trials   = 99
```

In []:

In [4]: `# data = channel 1, trial 2``data = EEGs[1,:,:2]``data.shape`Out[4]: `(640,)`

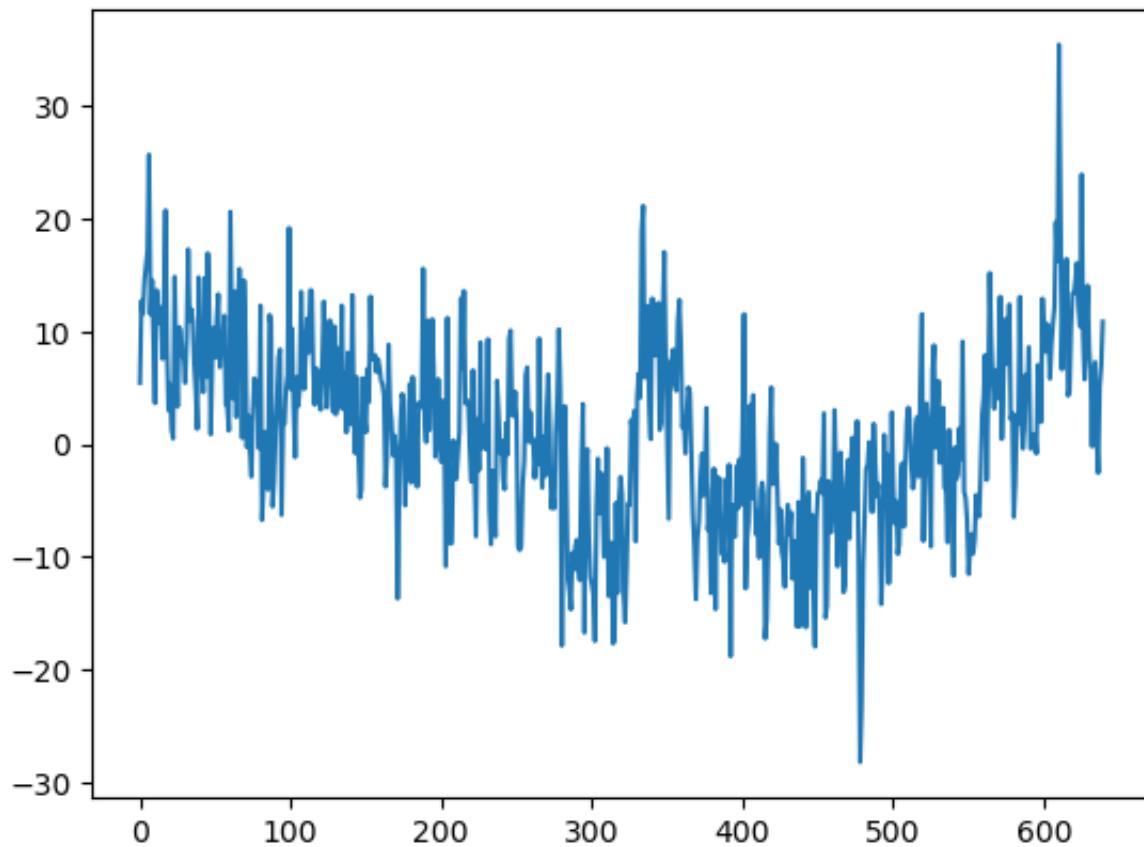
Let's plot this EEG.

In [5]: `import matplotlib.pyplot as plt``plt.plot(data)`Out[5]: `[<matplotlib.lines.Line2D at 0x10a5a6590>]`



Add `;` to the end of the last line to suppress the text output.

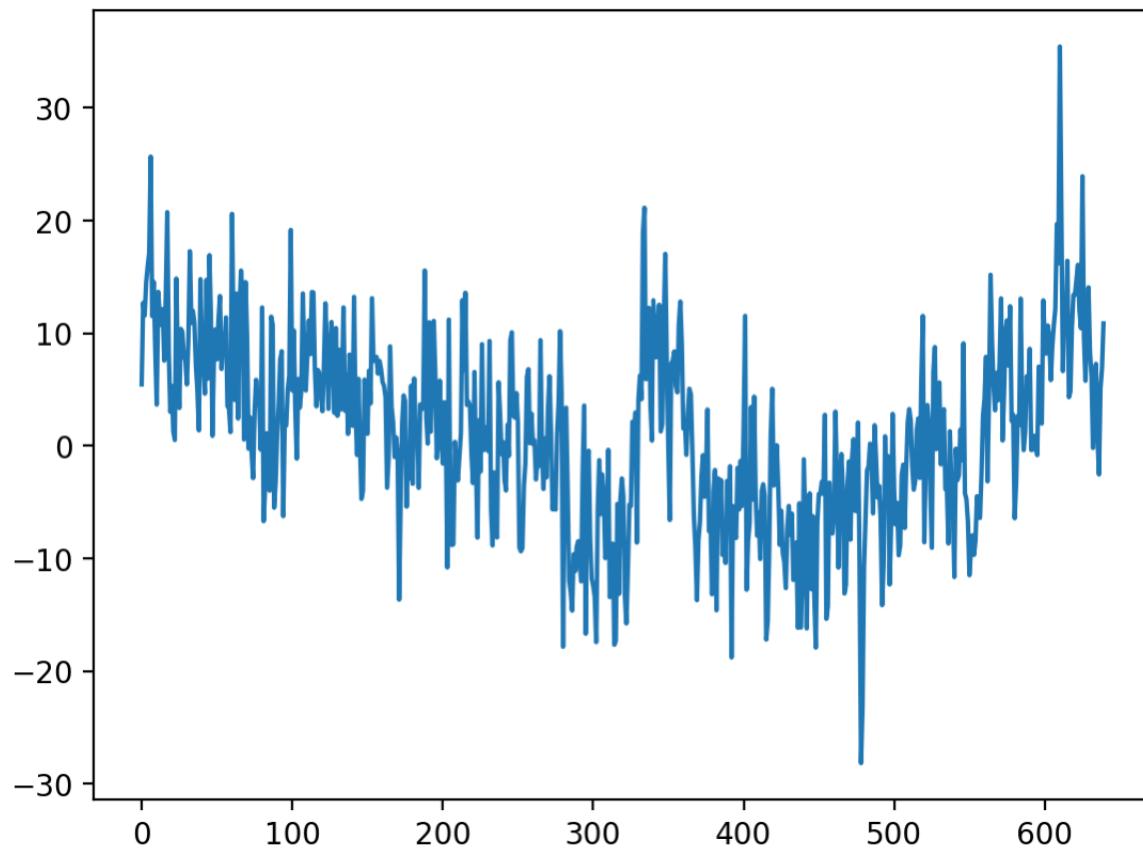
```
In [6]: plt.plot(data);
```



If you have a **retina display** (e.g., MacOS) this will make your plots look nicer.

```
In [7]: %config InlineBackend.figure_format = 'retina'
```

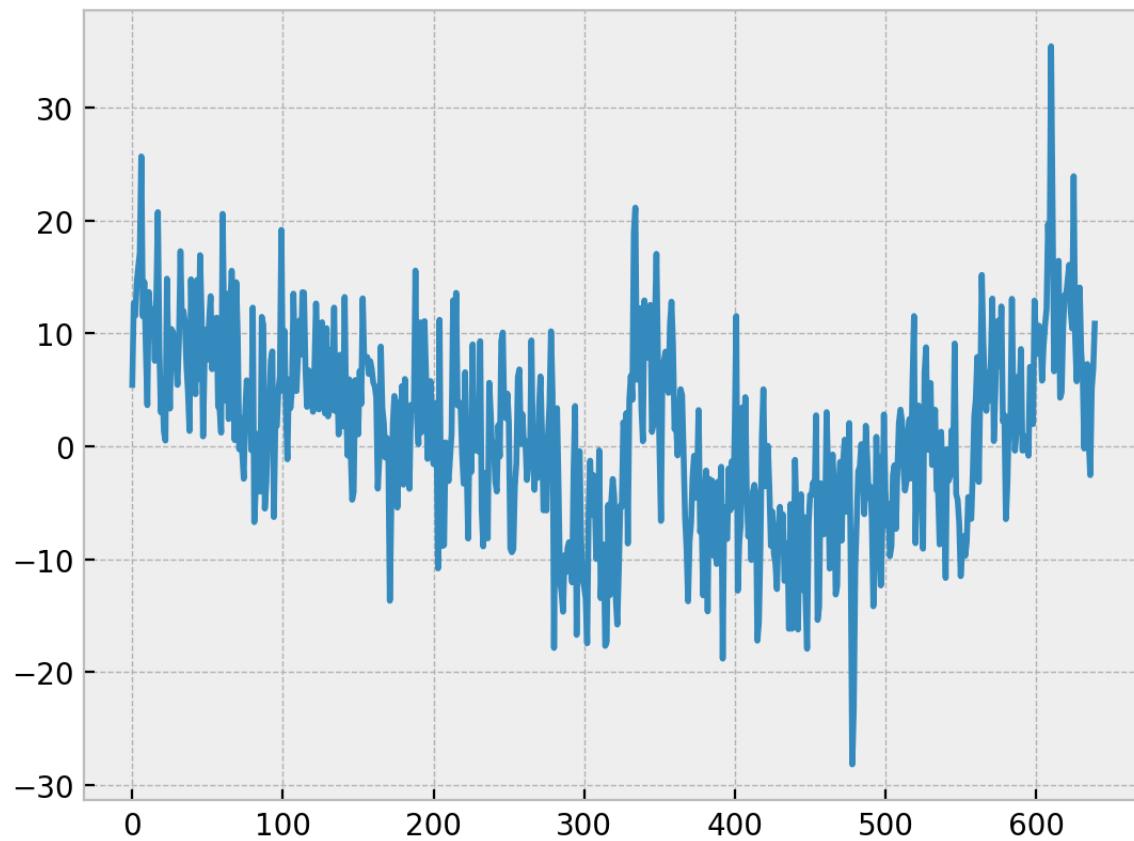
```
In [8]: plt.plot(data);
```



The default Matplotlib plot style is still not great. You can of course tweak all of the settings to get it just how you like it, but there are some preset styles you can load that are at least ok looking.

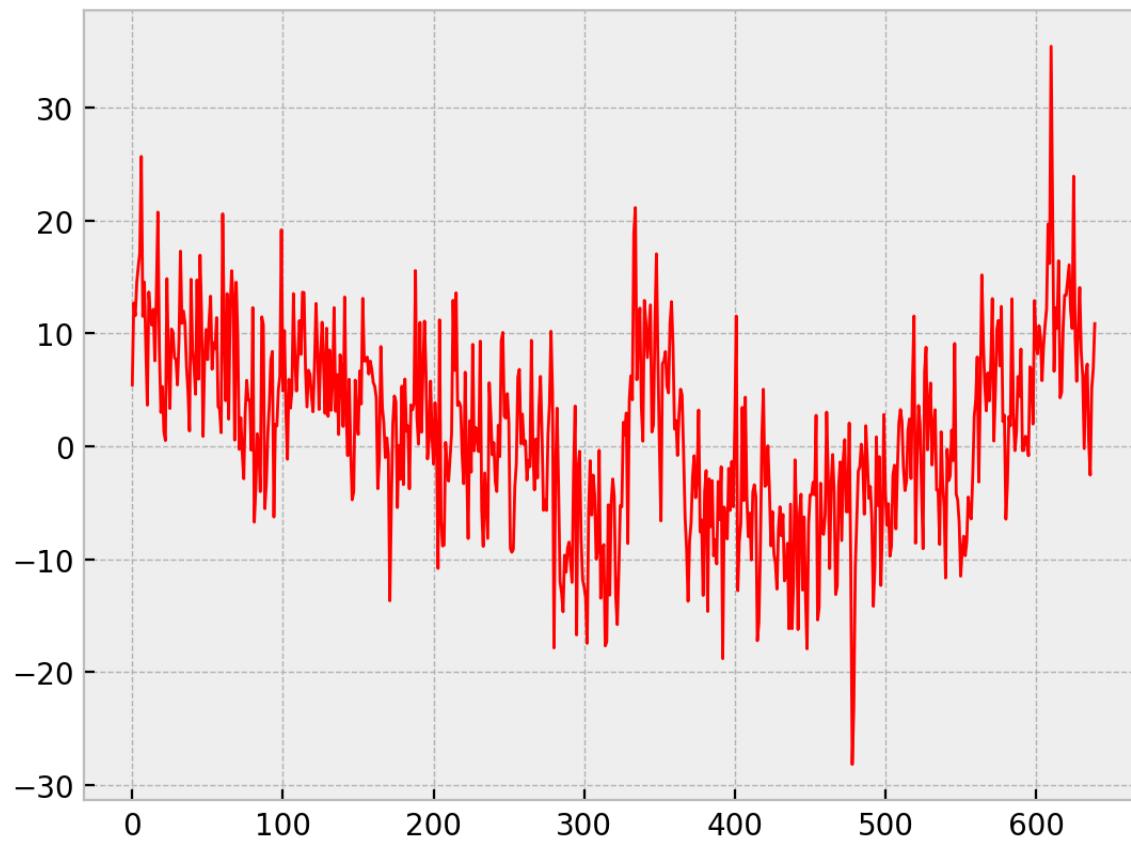
```
In [9]: plt.style.use("bmh")
```

```
In [10]: plt.plot(data);
```



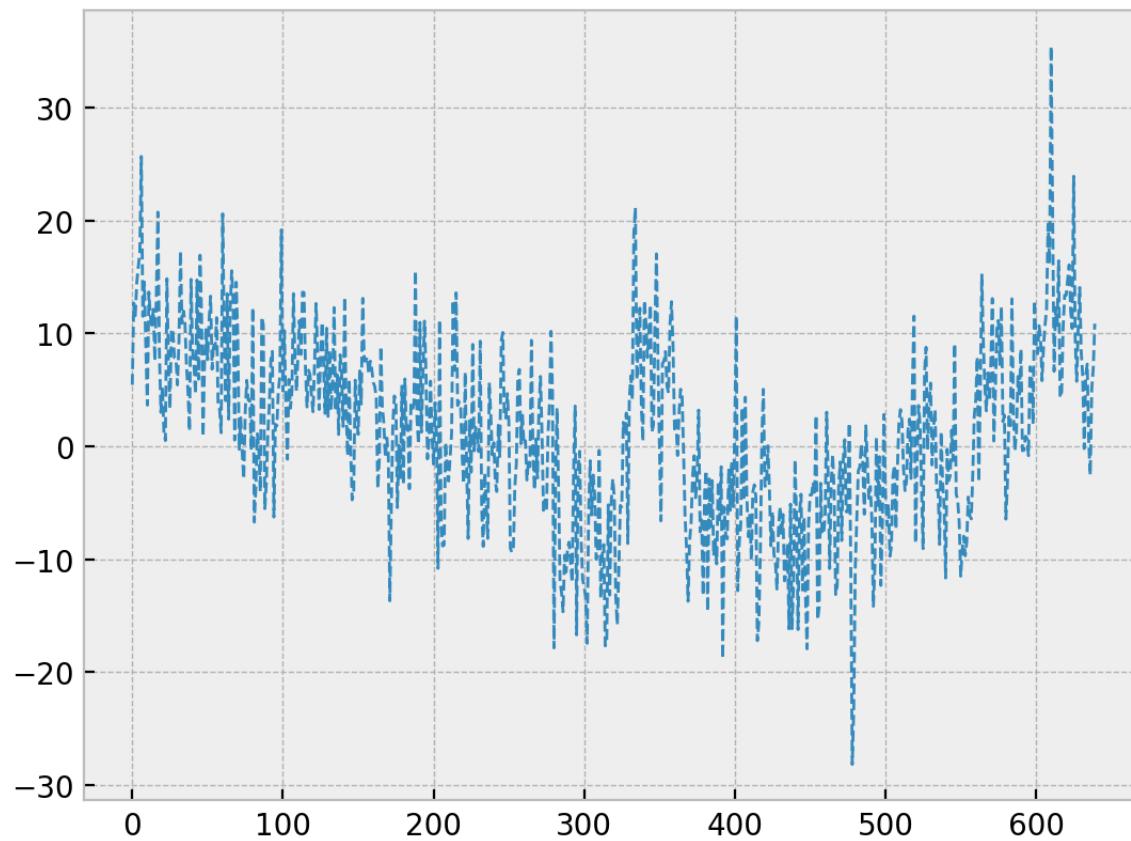
Specify line width using `lw=` and line color using `c=`

```
In [11]: plt.plot(data, lw=1, c='red');
```



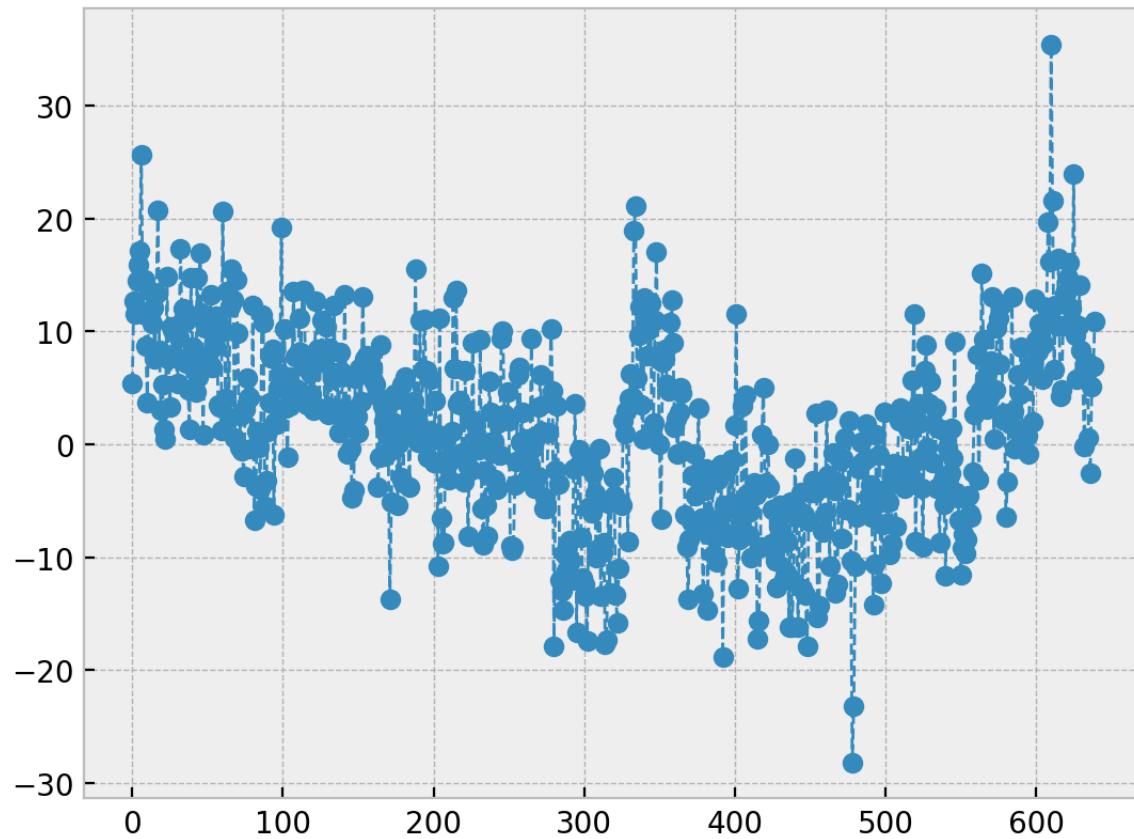
Specify a dashed line using `--`

```
In [12]: plt.plot(data, '--', lw=1);
```



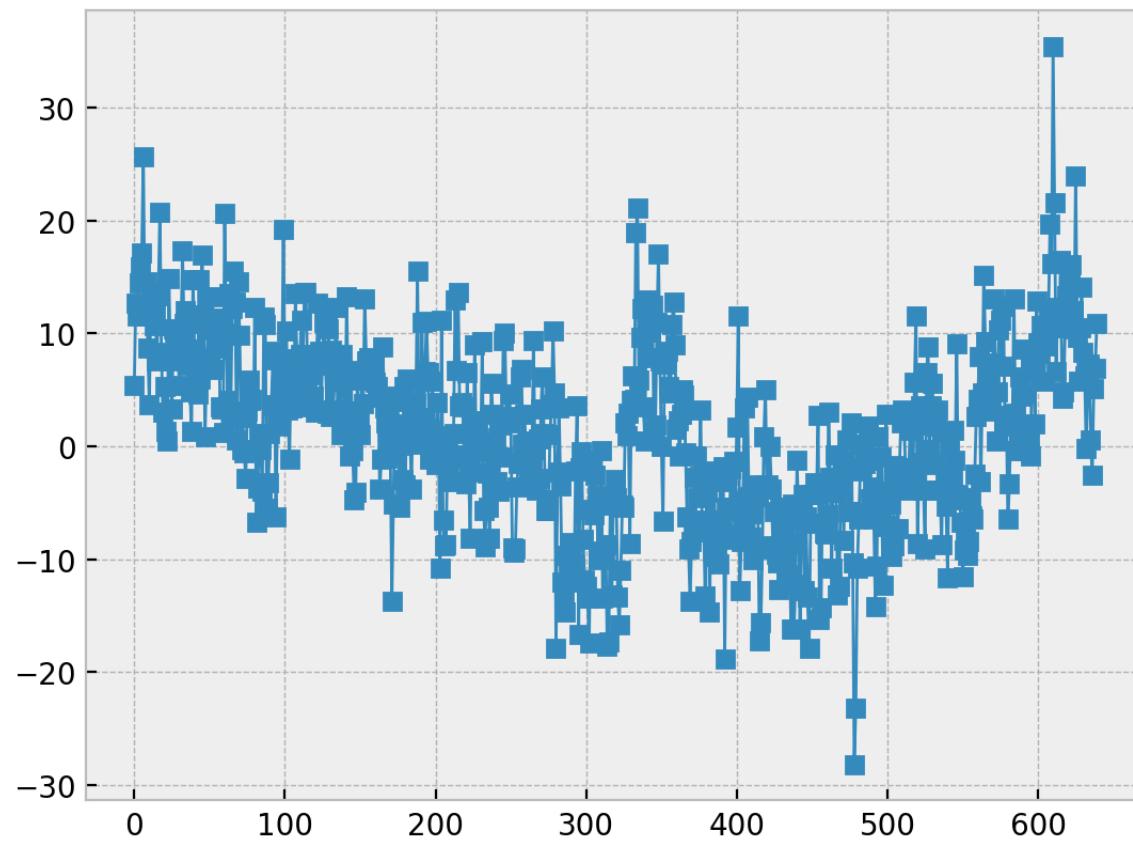
Specify round markers and a dashed line using `o--`

```
In [13]: plt.plot(data, 'o--', lw=1);
```



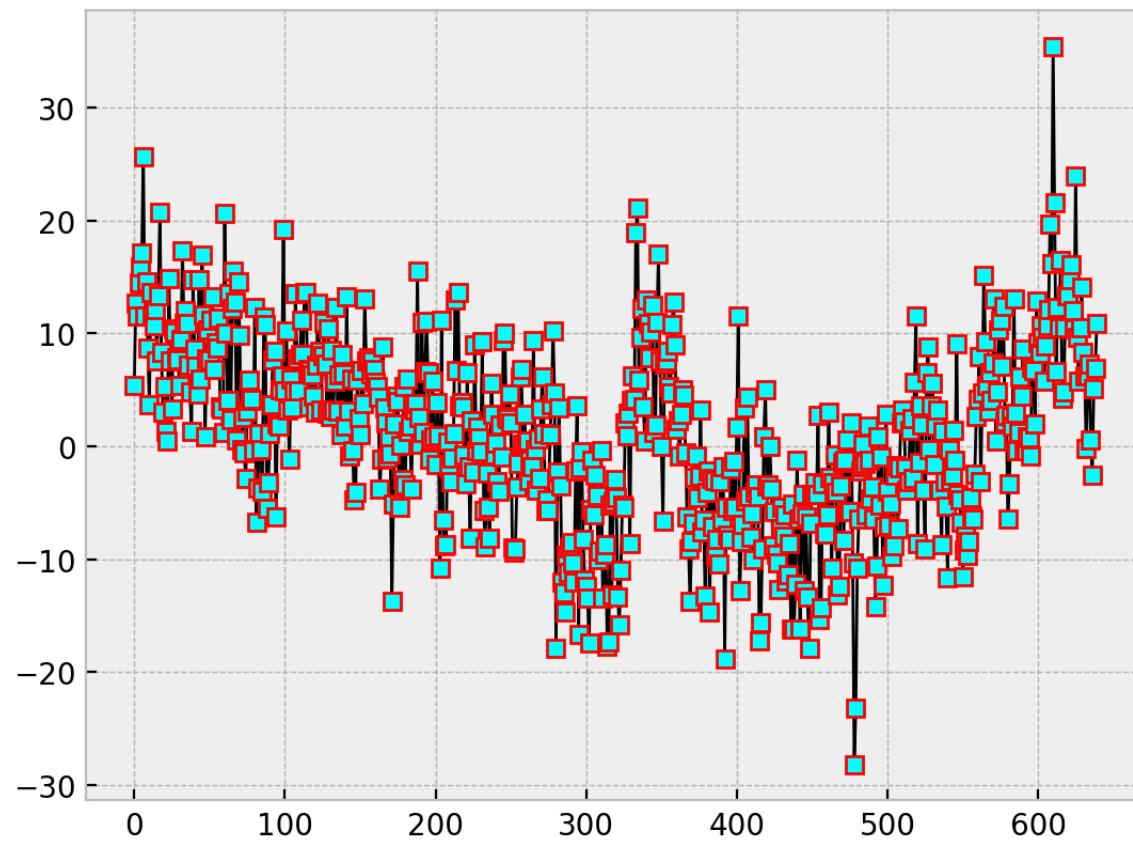
Specify square markers and a solid line using `s-`

In [14]: `plt.plot(data, 's-', lw=1);`



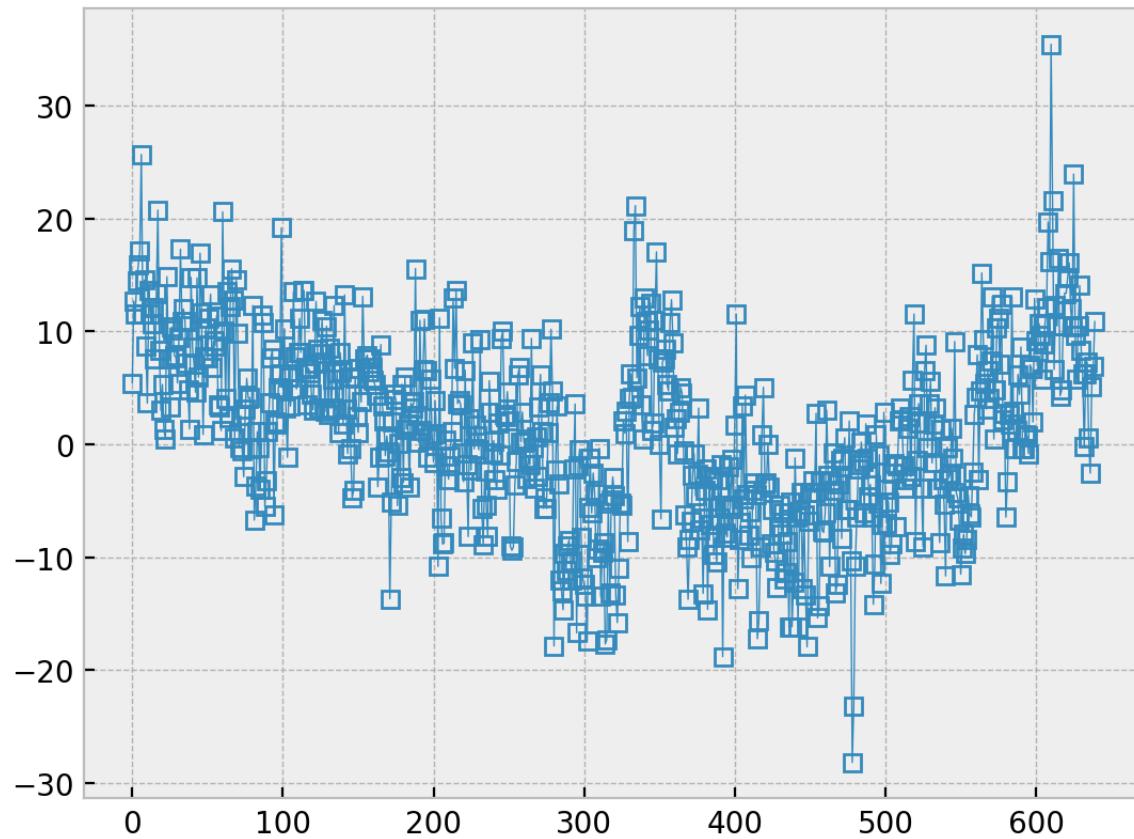
Specify marker edge and face colors with `mec=` and `mfc=`

```
In [15]: plt.plot(data, 's-', lw=1, c='black', mec='red', mfc=[0,1,1]);
```



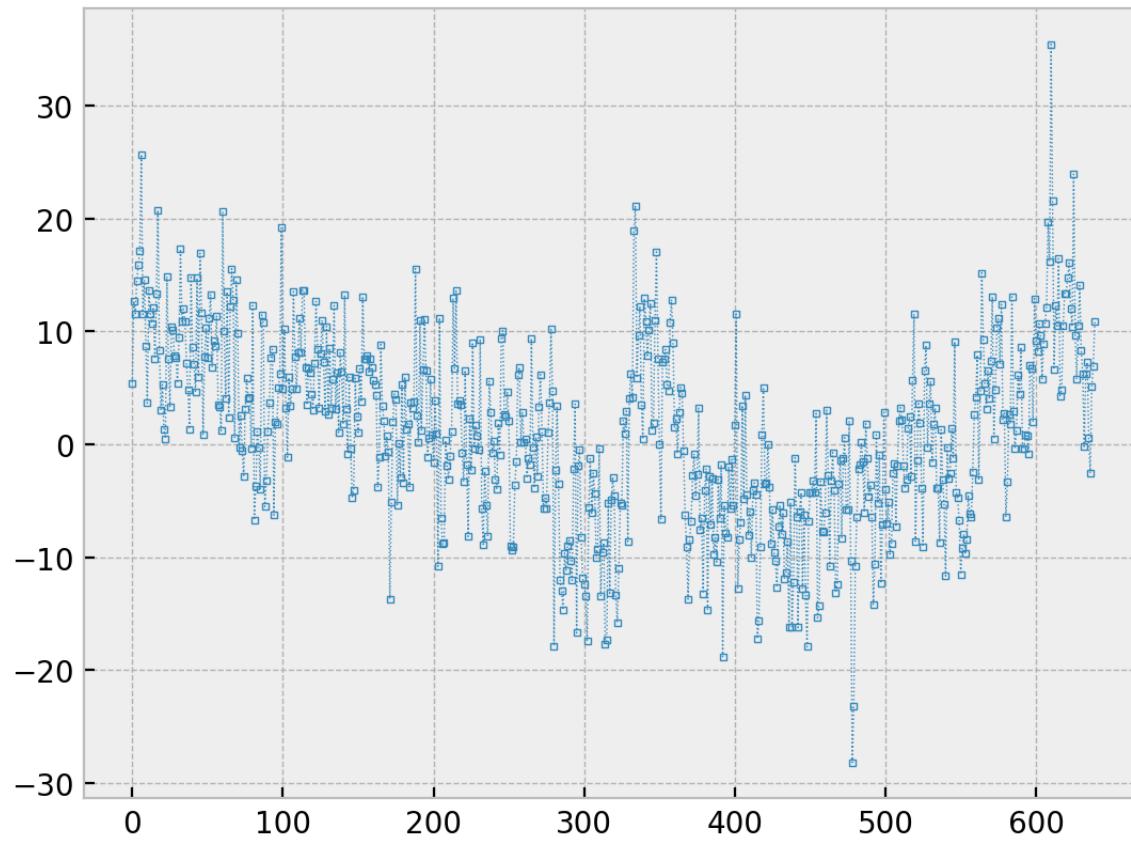
Specify non-filled markers with `mfc='none'`

```
In [16]: plt.plot(data, 's-', lw=0.5, mfc='none');
```



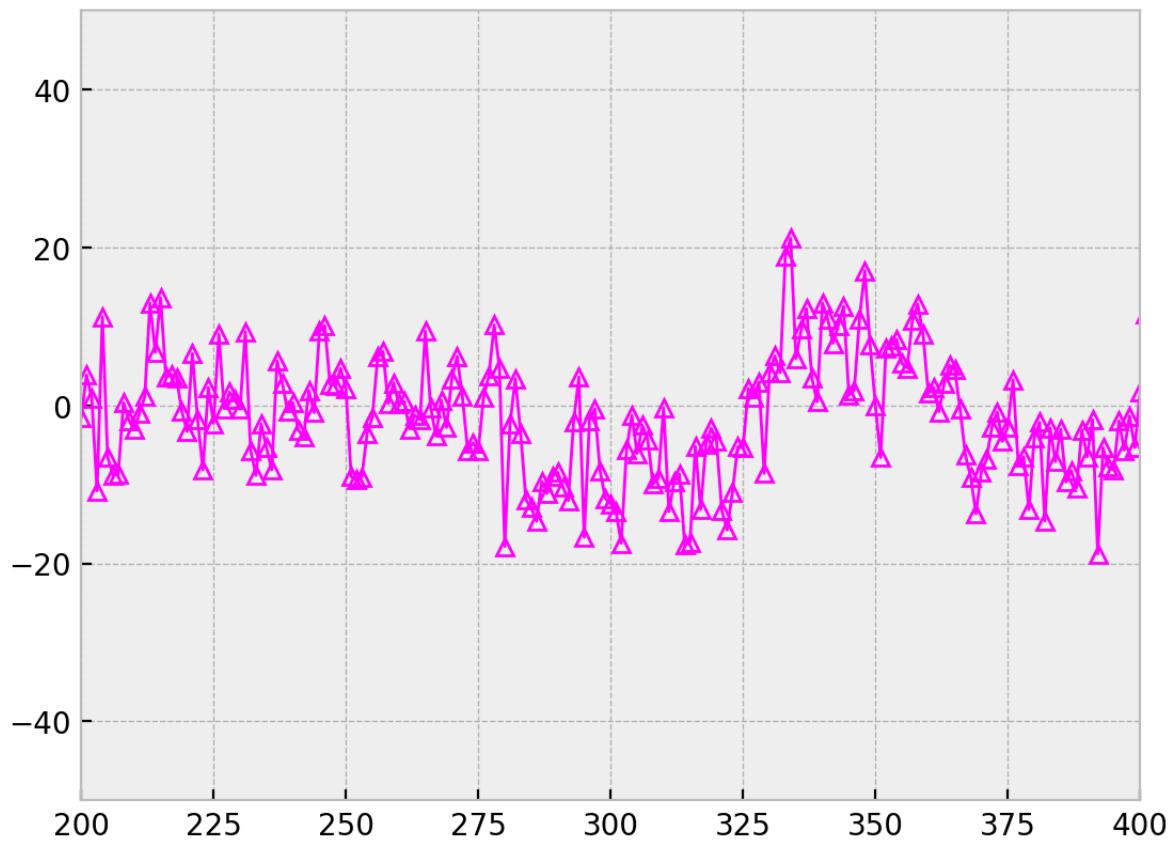
Specify marker edge width with `mew=` and marker size with `ms=`

```
In [17]: plt.plot(data, 's:', lw=0.5, mfc='none', mew=0.5, ms=2);
```



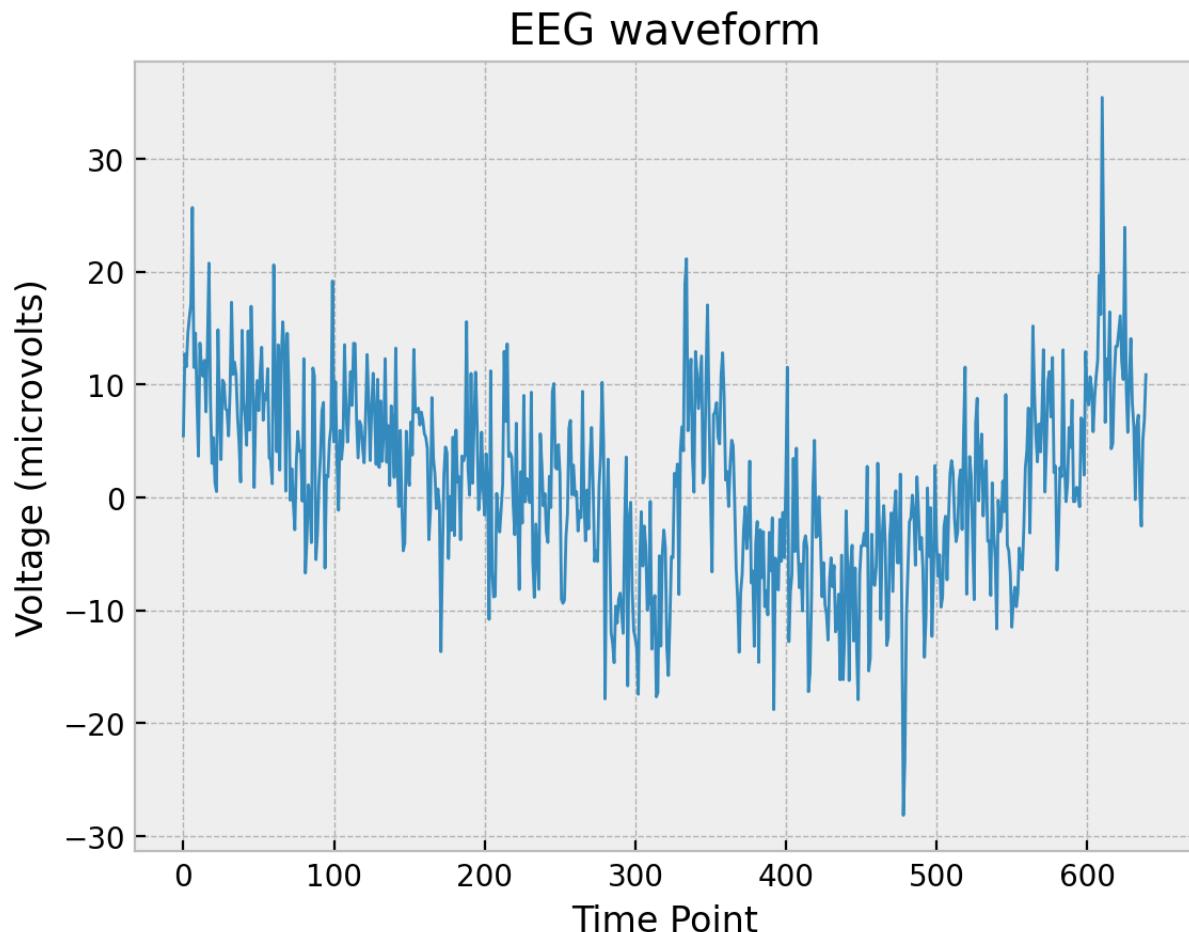
Set axes limits.

```
In [18]: plt.plot(data, '^-', lw=1, c='magenta', mfc='none')
plt.xlim([200, 400])
plt.ylim([-50, 50]);
```



Without **axes labels (including units)** your plot is **NOT COMPLETE!!!**

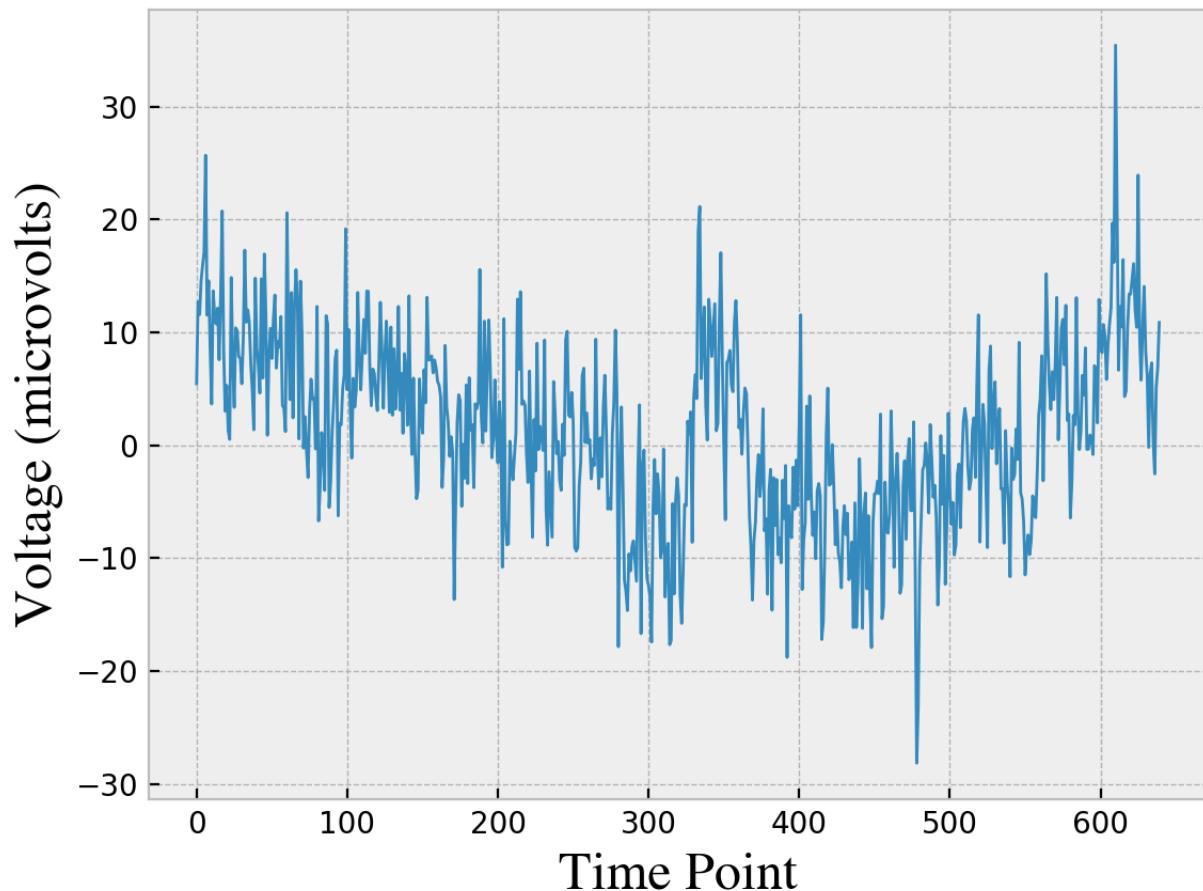
```
In [19]: plt.plot(data, lw=1)
plt.xlabel('Time Point')
plt.ylabel('Voltage (microvolts)')
plt.title('EEG waveform');
```



Specify axes label and plot title font, size and color.

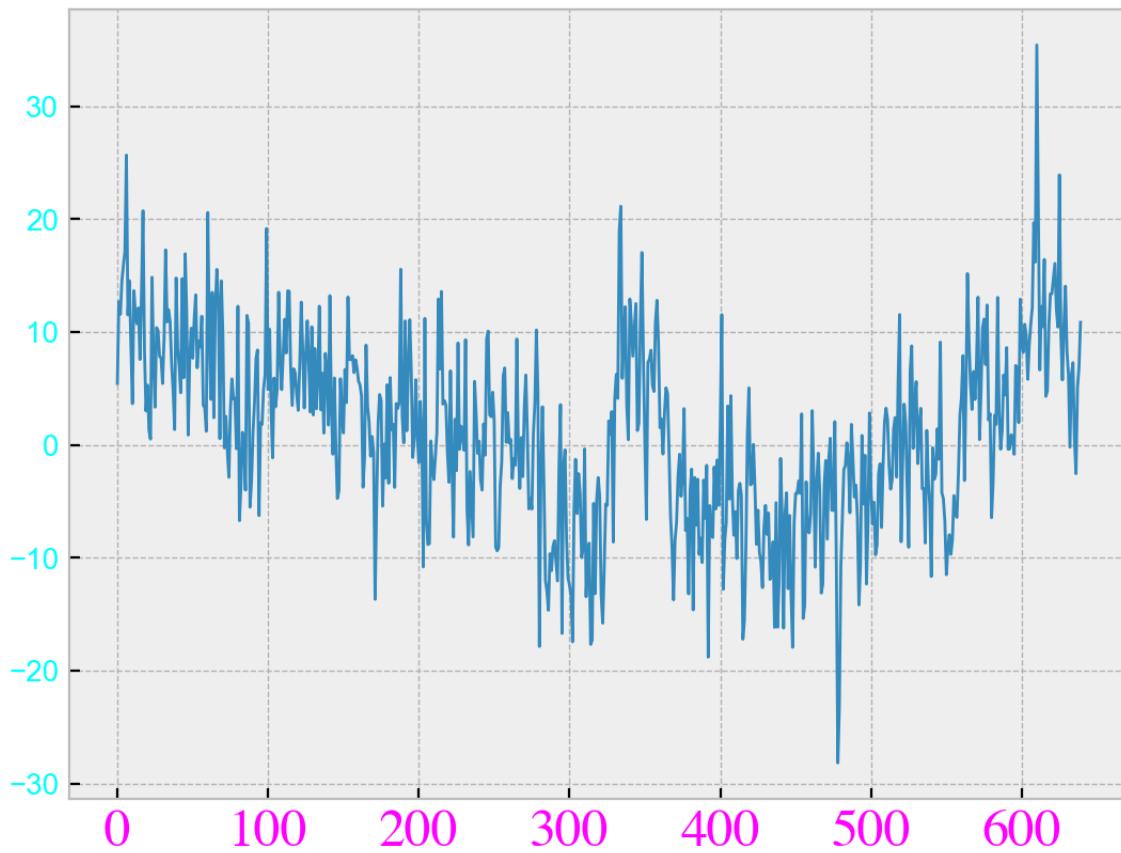
```
In [20]: plt.plot(data, lw=1)
plt.xlabel('Time Point', font='Times', size=18)
plt.ylabel('Voltage (microvolts)', font='Times', size=18)
plt.title('EEG waveform', font='Arial', size=24, color='red');
```

EEG waveform



Specify axes tick labels font, size and color.

```
In [21]: plt.plot(data, lw=1)
plt.xticks(font='Times', size=18, color=[1,0,1])
plt.yticks(font='Arial', size=10, color='cyan');
```

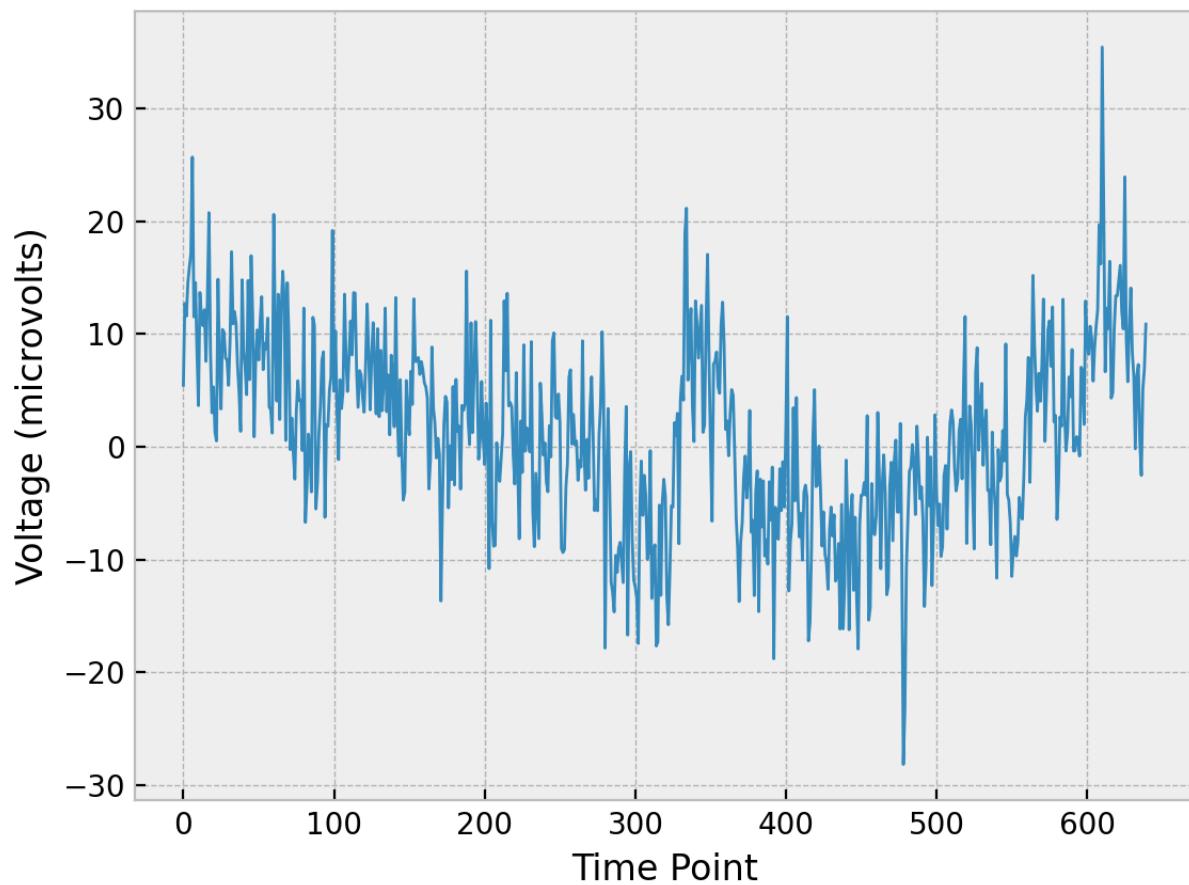


Instead of changing the title font size for each plot individually, let's set a new default size for all plots.

```
In [22]: plt.rcParams['axes.titlesize'] = 32  
plt.rcParams['lines.linewidth'] = 1
```

```
In [23]: plt.plot(data)  
plt.xlabel('Time Point')  
plt.ylabel('Voltage (microvolts)')  
plt.title('EEG waveform');
```

EEG waveform



How did I know what rcParams key to edit?

Well, rcParams is just a dictionary...

```
In [24]: plt.rcParams.keys()
```

```
Out[24]: KeysView(RcParams({'_internal.classic_mode': False,
                            'agg.path.chunkszie': 0,
                            'animation.bitrate': -1,
                            'animation.codec': 'h264',
                            'animation.convert_args': ['-layers', 'OptimizePlus'],
                            'animation.convert_path': 'convert',
                            'animation.embed_limit': 20.0,
                            'animation.ffmpeg_args': [],
                            'animation.ffmpeg_path': 'ffmpeg',
                            'animation.frame_format': 'png',
                            'animation.html': 'none',
                            'animation.writer': 'ffmpeg',
                            'axes.autolimit_mode': 'data',
                            'axes.axisbelow': 'line',
                            'axes.edgecolor': '#bcbcbc',
```

```
'axes.facecolor': '#eeeeee',
'axes.formatter.limits': [-5, 6],
'axes.formatter.min_exponent': 0,
'axes.formatter.offset_threshold': 4,
'axes.formatter.use_locale': False,
'axes.formatter.use_mathtext': False,
'axes.formatter.useoffset': True,
'axes.grid': True,
'axes.grid.axis': 'both',
'axes.grid.which': 'major',
'axes.labelcolor': 'black',
'axes.labelpad': 4.0,
'axes.labelsize': 'large',
'axes.labelweight': 'normal',
'axes.linewidth': 0.8,
'axes.prop_cycle': cycler('color', ['#348ABD', '#A60628', '#7A68A6', '#467821', '#D55E00', '#CC79A7', '#56B4E9', '#009E73', '#F0E442', '#0072B2']),
'axes.spines.bottom': True,
'axes.spines.left': True,
'axes.spines.right': True,
'axes.spines.top': True,
'axes.titlecolor': 'auto',
'axes.titlelocation': 'center',
'axes.titlepad': 6.0,
'axes.titlesize': 32.0,
'axes.titleweight': 'normal',
'axes.titley': None,
'axes.unicode_minus': True,
'axes.xmargin': 0.05,
'axes.ymargin': 0.05,
'axes.zmargin': 0.05,
'axes3d.grid': True,
'backend': 'module://matplotlib_inline.backend_inline',
'backend_fallback': True,
'boxplot.bootstrap': None,
'boxplot.boxprops.color': 'black',
'boxplot.boxprops.linestyle': '-',
'boxplot.boxprops.linewidth': 1.0,
'boxplot.capprops.color': 'black',
'boxplot.capprops.linestyle': '-',
'boxplot.capprops.linewidth': 1.0,
'boxplot.flierprops.color': 'black',
'boxplot.flierprops.linestyle': 'none',
'boxplot.flierprops.linewidth': 1.0,
'boxplot.flierprops.marker': 'o',
'boxplot.flierprops.markeredgecolor': 'black',
'boxplot.flierprops.markeredgewidth': 1.0,
'boxplot.flierprops.markerfacecolor': 'none',
'boxplot.flierprops.markersize': 6.0,
```

```
'boxplot.meanline': False,  
'boxplot.meanprops.color': 'C2',  
'boxplot.meanprops.linestyle': '--',  
'boxplot.meanprops.linewidth': 1.0,  
'boxplot.meanprops.marker': '^',  
'boxplot.meanprops.markeredgecolor': 'C2',  
'boxplot.meanprops.markerfacecolor': 'C2',  
'boxplot.meanprops.markersize': 6.0,  
'boxplot.medianprops.color': 'C1',  
'boxplot.medianprops.linestyle': '-',  
'boxplot.medianprops.linewidth': 1.0,  
'boxplot.notch': False,  
'boxplot.patchartist': False,  
'boxplot.showbox': True,  
'boxplot.showcaps': True,  
'boxplot.showfliers': True,  
'boxplot.showmeans': False,  
'boxplot.vertical': True,  
'boxplot.whiskerprops.color': 'black',  
'boxplot.whiskerprops.linestyle': '-',  
'boxplot.whiskerprops.linewidth': 1.0,  
'boxplot.whiskers': 1.5,  
'contour.algorithm': 'mpl2014',  
'contour.corner_mask': True,  
'contour.linewidth': None,  
'contour.negative_linestyle': 'dashed',  
'date.autoformatter.day': '%Y-%m-%d',  
'date.autoformatter.hour': '%m-%d %H',  
'date.autoformatter.microsecond': '%M:%S.%f',  
'date.autoformatter.minute': '%d %H:%M',  
'date.autoformatter.month': '%Y-%m',  
'date.autoformatter.second': '%H:%M:%S',  
'date.autoformatter.year': '%Y',  
'date.converter': 'auto',  
'date.epoch': '1970-01-01T00:00:00',  
'date.interval_multiples': True,  
'docstring.hardcopy': False,  
'errorbar.capsize': 0.0,  
'figure.autolayout': False,  
'figure.constrained_layout.h_pad': 0.04167,  
'figure.constrained_layout.hspace': 0.02,  
'figure.constrained_layout.use': False,  
'figure.constrained_layout.w_pad': 0.04167,  
'figure.constrained_layout.wspace': 0.02,  
'figure.dpi': 100.0,  
'figure.edgecolor': 'white',  
'figure.facecolor': 'white',  
'figure.figsize': [6.4, 4.8],  
'figure.frameon': True,  
'figure.labelsizer': 'large',
```

```
'figure.labelweight': 'normal',
'figure.max_open_warning': 20,
'figure.raise_window': True,
'figure.subplot.bottom': 0.11,
'figure.subplot.hspace': 0.2,
'figure.subplot.left': 0.125,
'figure.subplot.right': 0.9,
'figure.subplot.top': 0.88,
'figure.subplot.wspace': 0.2,
'figure.titlesize': 'large',
'figure.titleweight': 'normal',
'font.cursive': ['Apple Chancery',
                 'Textile',
                 'Zapf Chancery',
                 'Sand',
                 'Script MT',
                 'Felipa',
                 'Comic Neue',
                 'Comic Sans MS',
                 'cursive'],
'font.family': ['sans-serif'],
'font.fantasy': ['Chicago',
                 'Charcoal',
                 'Impact',
                 'Western',
                 'Humor Sans',
                 'xkcd',
                 'fantasy'],
'font.monospace': ['DejaVu Sans Mono',
                   'Bitstream Vera Sans Mono',
                   'Computer Modern Typewriter',
                   'Andale Mono',
                   'Nimbus Mono L',
                   'Courier New',
                   'Courier',
                   'Fixed',
                   'Terminal',
                   'monospace'],
'font.sans-serif': ['DejaVu Sans',
                    'Bitstream Vera Sans',
                    'Computer Modern Sans Serif',
                    'Lucida Grande',
                    'Verdana',
                    'Geneva',
                    'Lucid',
                    'Arial',
                    'Helvetica',
                    'Avant Garde',
                    'sans-serif'],
'font.serif': ['DejaVu Serif',
```

```
'Bitstream Vera Serif',
'Computer Modern Roman',
'New Century Schoolbook',
'Century Schoolbook L',
'Utopia',
'ITC Bookman',
'Bookman',
'Nimbus Roman No9 L',
'Times New Roman',
'Times',
'Palatino',
'Charter',
'serif'],
'font.size': 10.0,
'font.stretch': 'normal',
'font.style': 'normal',
'font.variant': 'normal',
'font.weight': 'normal',
'grid.alpha': 1.0,
'grid.color': '#b2b2b2',
'grid.linestyle': '--',
'grid.linewidth': 0.5,
'hatch.color': 'black',
'hatch.linewidth': 1.0,
'hist.bins': 10,
'image.aspect': 'equal',
'image.cmap': 'viridis',
'image.composite_image': True,
'image.interpolation': 'antialiased',
'image.lut': 256,
'image.origin': 'upper',
'image.resample': True,
'interactive': True,
'keymap.back': ['left', 'c', 'backspace', 'MouseButton.BACK'],
'keymap.copy': ['ctrl+c', 'cmd+c'],
'keymap.forward': ['right', 'v', 'MouseButton.FORWARD'],
'keymap.fullscreen': ['f', 'ctrl+f'],
'keymap.grid': ['g'],
'keymap.grid_minor': ['G'],
'keymap.help': ['f1'],
'keymap.home': ['h', 'r', 'home'],
'keymap.pan': ['p'],
'keymap.quit': ['ctrl+w', 'cmd+w', 'q'],
'keymap.quit_all': [],
'keymap.save': ['s', 'ctrl+s'],
'keymap.xscale': ['k', 'L'],
'keymap.yscale': ['l'],
'keymap.zoom': ['o'],
'legend.borderaxespad': 0.5,
'legend.borderpad': 0.4,
```

```
'legend.columnspacing': 2.0,
'legend.edgecolor': '0.8',
'legend.facecolor': 'inherit',
'legend.fancybox': True,
'legend.fontsize': 'medium',
'legend.framealpha': 0.8,
'legend.frameon': True,
'legend.handleheight': 0.7,
'legend.handlelength': 2.0,
'legend.handletextpad': 0.8,
'legend.labelcolor': 'None',
'legend.labelsspacing': 0.5,
'legend.loc': 'best',
'legend.markerscale': 1.0,
'legend.numpoints': 1,
'legend.scatterpoints': 1,
'legend.shadow': False,
'legend.title_fontsize': None,
'lines.antialiased': True,
'lines.color': 'C0',
'lines.dash_capstyle': <CapStyle.butt: 'butt'>,
'lines.dash_joinstyle': <JoinStyle.round: 'round'>,
'lines.dashdot_pattern': [6.4, 1.6, 1.0, 1.6],
'lines.dashed_pattern': [3.7, 1.6],
'lines.dotted_pattern': [1.0, 1.65],
'lines.linestyle': '-',
'lines.linewidth': 1.0,
'lines.marker': 'None',
'lines.markeredgecolor': 'auto',
'lines.markeredgewidth': 1.0,
'lines.markerfacecolor': 'auto',
'lines.markersize': 6.0,
'lines.scale_dashes': True,
'lines.solid_capstyle': <CapStyle.projecting: 'projecting'>,
'lines.solid_joinstyle': <JoinStyle.round: 'round'>,
'markers.fillstyle': 'full',
'mathtext.bf': 'sans:bold',
'mathtext.cal': 'cursive',
'mathtext.default': 'it',
'mathtext.fallback': 'cm',
'mathtext.fontset': 'cm',
'mathtext.it': 'sans:italic',
'mathtext.rm': 'sans',
'mathtext.sf': 'sans',
'mathtext.tt': 'monospace',
'patch.antialiased': True,
'patch.edgecolor': '#eeeeee',
'patch.facecolor': 'blue',
'patch.force_edgecolor': False,
'patch.linewidth': 0.5,
```

```
'path.effects': [],
'path.simplify': True,
'path.simplify_threshold': 0.111111111111,
'path.sketch': None,
'path.snap': True,
'pcolor.shading': 'auto',
'pcolormesh.snap': True,
'pdf.compression': 6,
'pdf.fonttype': 3,
'pdf.inheritcolor': False,
'pdf.use14corefonts': False,
'pgf.preamble': '',
'pgf.rcfonts': True,
'pgf.texsystem': 'xelatex',
'polaraxes.grid': True,
'ps.distiller.res': 6000,
'ps.fonttype': 3,
'ps.paperize': 'letter',
'ps.useafm': False,
'ps.usedistiller': None,
'savefig.bbox': None,
'savefig.directory': '~',
'savefig.dpi': 'figure',
'savefig.edgecolor': 'auto',
'savefig.facecolor': 'auto',
'savefig.format': 'png',
'savefig.orientation': 'portrait',
'savefig.pad_inches': 0.1,
'savefig.transparent': False,
'scatter.edgecolors': 'face',
'scatter.marker': 'o',
'svg.fonttype': 'path',
'svg.hashsalt': None,
'svg.image_inline': True,
'text.antialiased': True,
'text.color': 'black',
'text.hinting': 'force_autohint',
'text.hinting_factor': 8,
'text.kerning_factor': 0,
'text.latex.preamble': '',
'text.parse_math': True,
'text.usetex': False,
'timezone': 'UTC',
'tk.window_focus': False,
'toolbar': 'toolbar2',
'webagg.address': '127.0.0.1',
'webagg.open_in_browser': True,
'webagg.port': 8988,
'webagg.port_retries': 50,
'xaxis.labellocation': 'center',
```

```
'xtick.alignment': 'center',
'xtick.bottom': True,
'xtick.color': 'black',
'xtick.direction': 'in',
'xtick.labelbottom': True,
'xtick.labelcolor': 'inherit',
'xtick.labelsizes': 'medium',
'xtick.labeltop': False,
'xtick.major.bottom': True,
'xtick.major.pad': 3.5,
'xtick.major.size': 3.5,
'xtick.major.top': True,
'xtick.major.width': 0.8,
'xtick.minor.bottom': True,
'xtick.minor.pad': 3.4,
'xtick.minor.size': 2.0,
'xtick.minor.top': True,
'xtick.minor.visible': False,
'xtick.minor.width': 0.6,
'xtick.top': False,
'yaxis.labellocation': 'center',
'ytick.alignment': 'center_baseline',
'ytick.color': 'black',
'ytick.direction': 'in',
'ytick.labelcolor': 'inherit',
'ytick.labelleft': True,
'ytick.labelright': False,
'ytick.labelsizes': 'medium',
'ytick.left': True,
'ytick.major.left': True,
'ytick.major.pad': 3.5,
'ytick.major.right': True,
'ytick.major.size': 3.5,
'ytick.major.width': 0.8,
'ytick.minor.left': True,
'ytick.minor.pad': 3.4,
'ytick.minor.right': True,
'ytick.minor.size': 2.0,
'ytick.minor.visible': False,
'ytick.minor.width': 0.6,
'ytick.right': False}))
```

Pretty much everything is customizable, just google it.

But you can probably get most of the way there with just a few commands that you would typically put near the beginning of your notebook such as:

```
In [25]: import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
```

```
plt.style.use("bmh")
plt.rcParams['lines.linewidth'] = 1
```

Plot X vs. Y

```
In [26]: N = len(data)

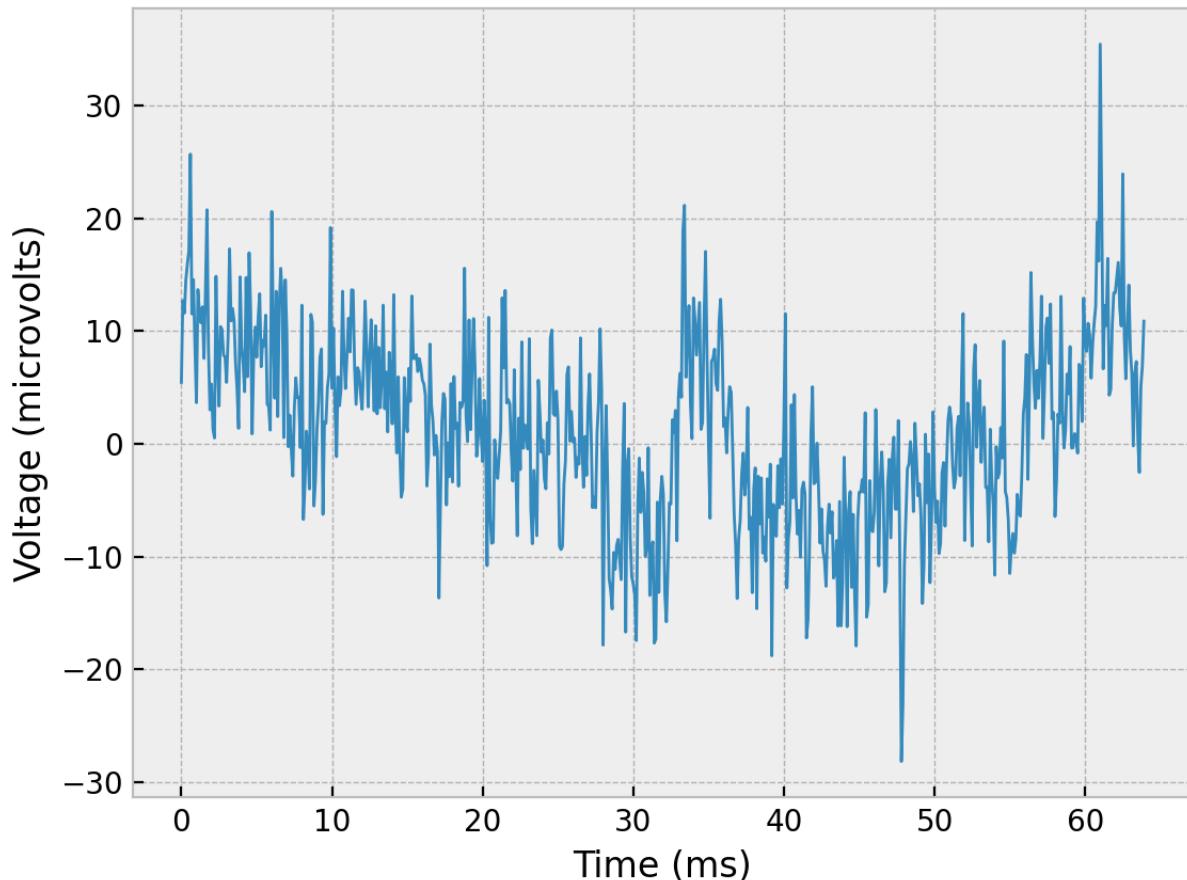
dt = 0.1 # sample interval in ms

# sample times (ms)
time = np.arange(N) * dt

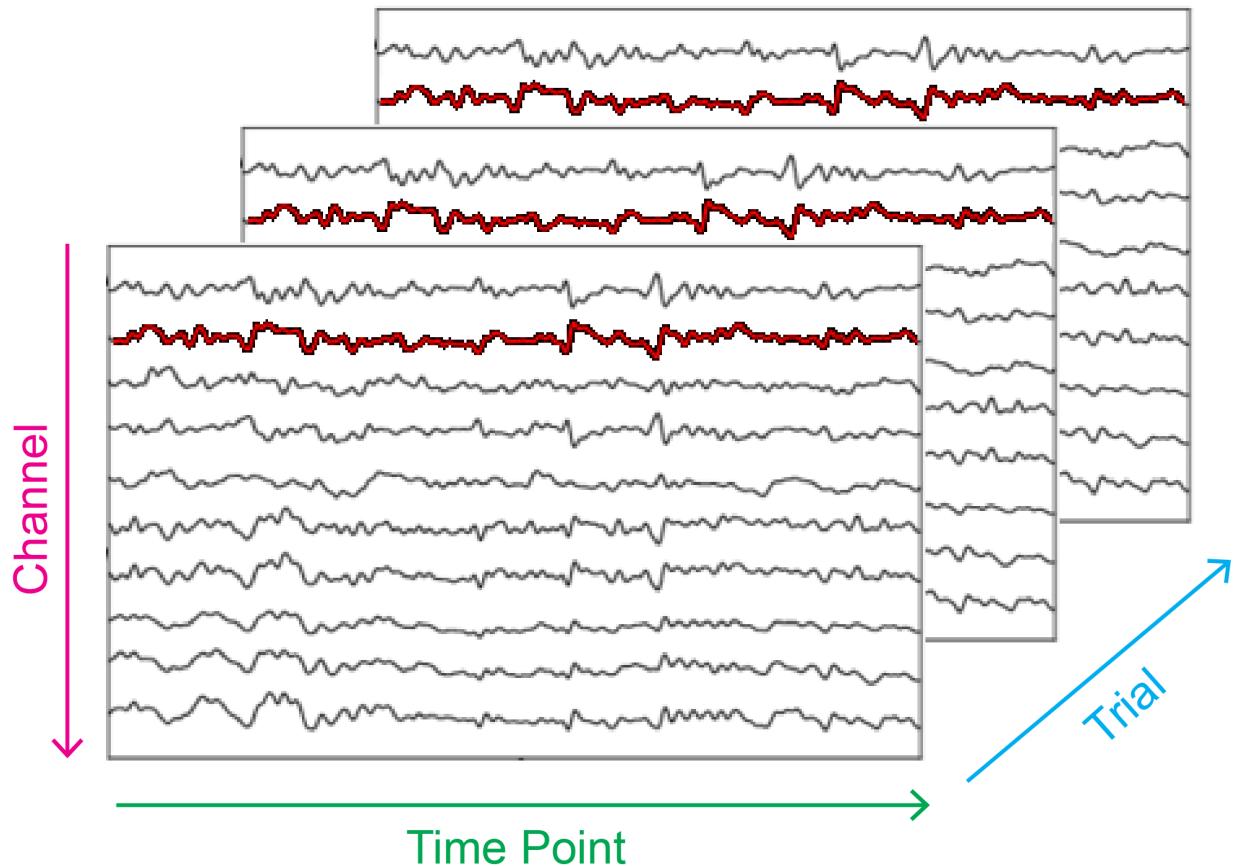
time.shape, data.shape
```

Out[26]: ((640,), (640,))

```
In [27]: plt.plot(time, data)
plt.xlabel('Time (ms)')
plt.ylabel('Voltage (microvolts)');
```



Plot the EEGs from channel 1 for the first three trials all together on the same axes.

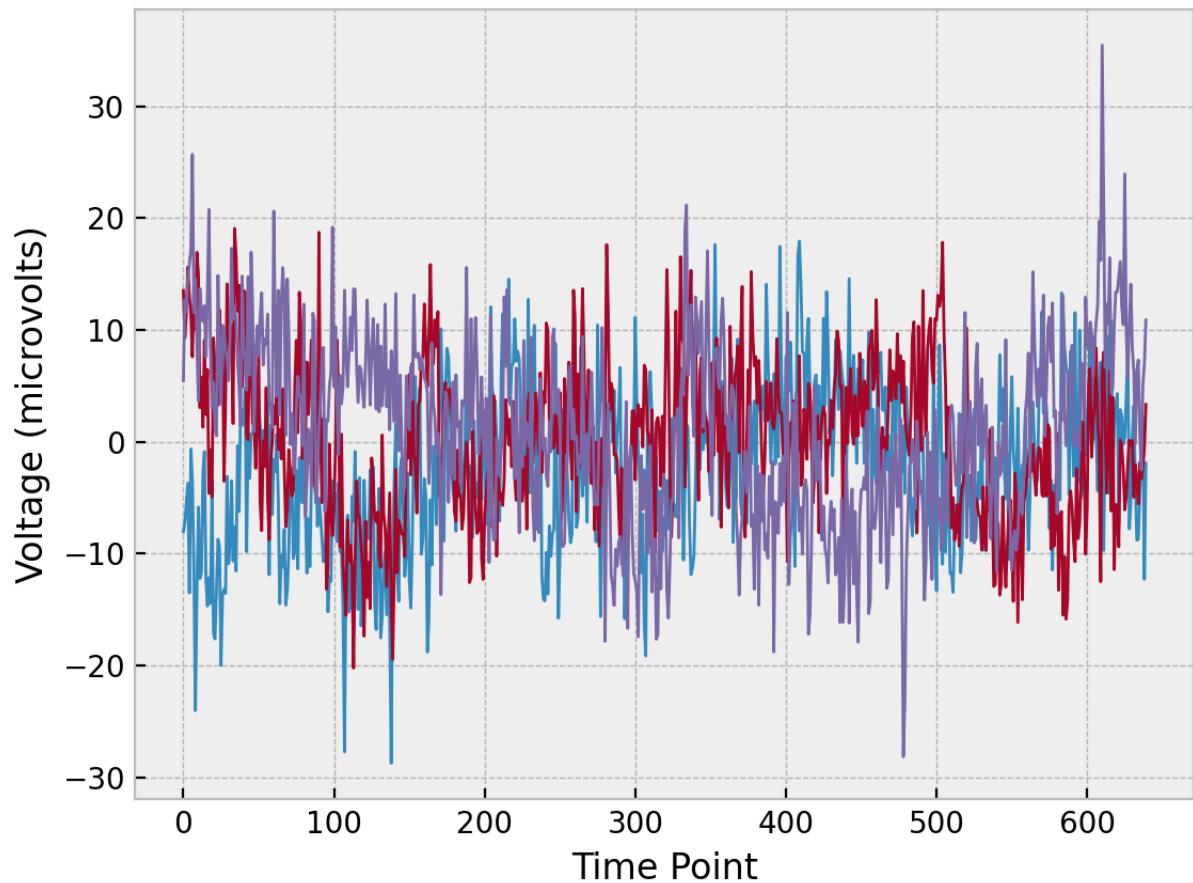


```
In [28]: # data = channel 1, first three trials  
  
data = EEGs[1,:,:3]  
  
data.shape
```

Out[28]: (640, 3)

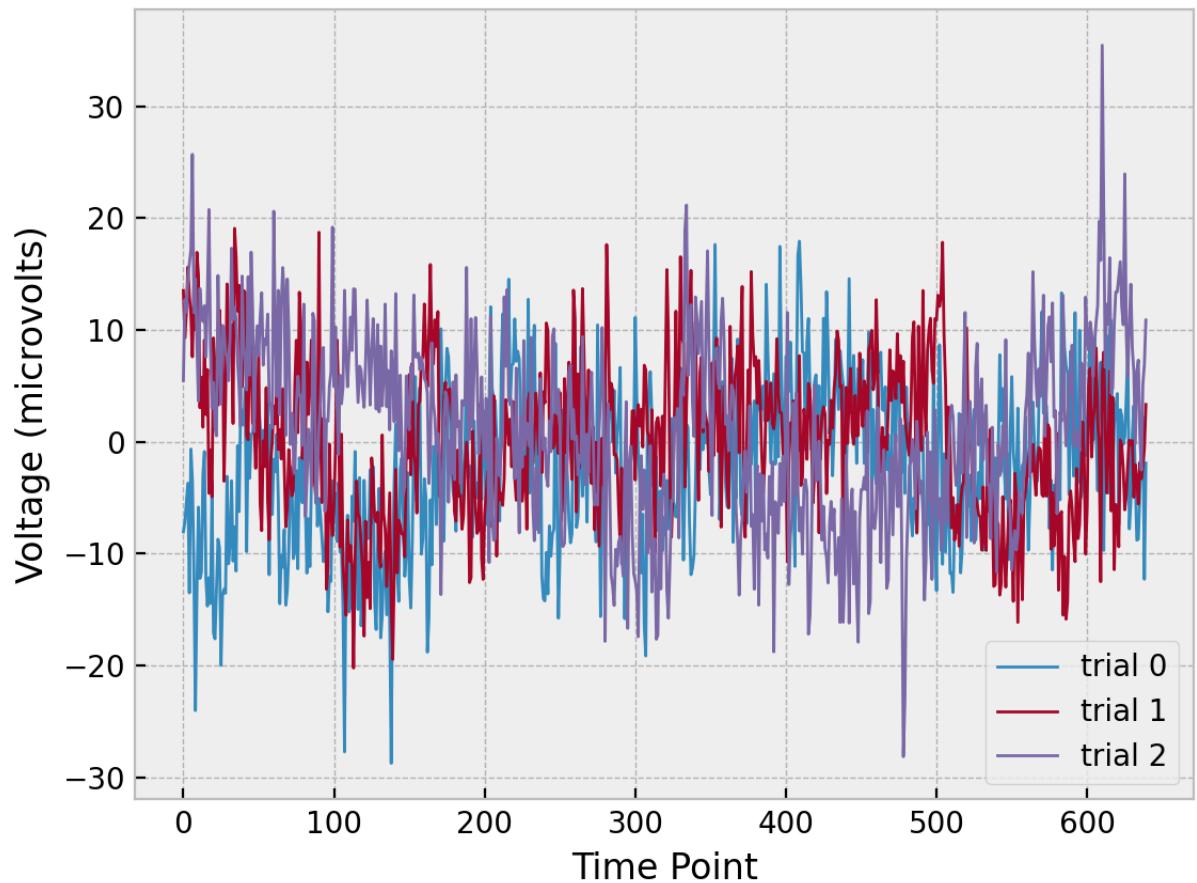
data now has the format data[time, trial]

```
In [29]: plt.plot(data[:,0]) # all time pts, trial 0  
plt.plot(data[:,1]) # all time pts, trial 1  
plt.plot(data[:,2]) # all time pts, trial 2  
plt.xlabel('Time Point')  
plt.ylabel('Voltage (microvolts)');
```



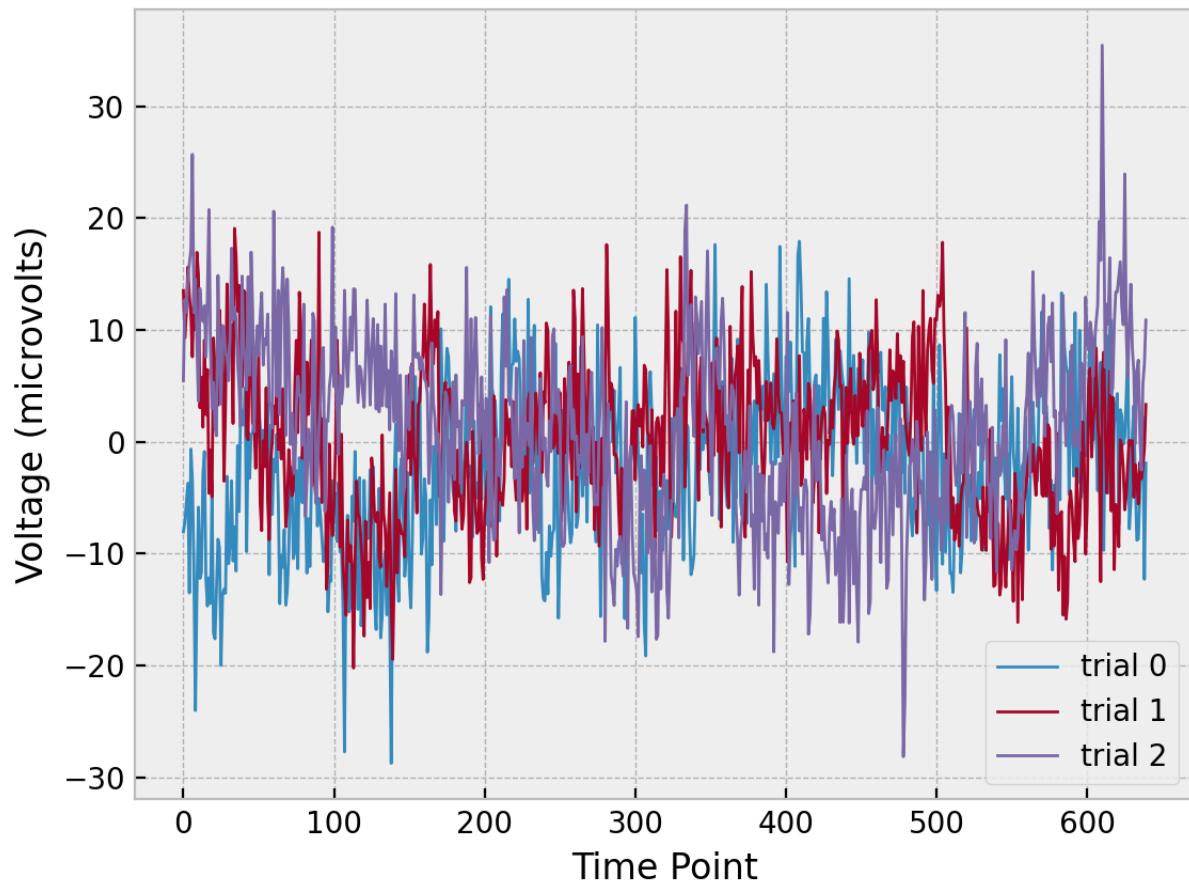
Add a legend and label each trace.

```
In [30]: plt.plot(data[:,0], label="trial 0")
plt.plot(data[:,1], label="trial 1")
plt.plot(data[:,2], label="trial 2")
plt.xlabel('Time Point')
plt.ylabel('Voltage (microvolts)')
plt.legend();
```



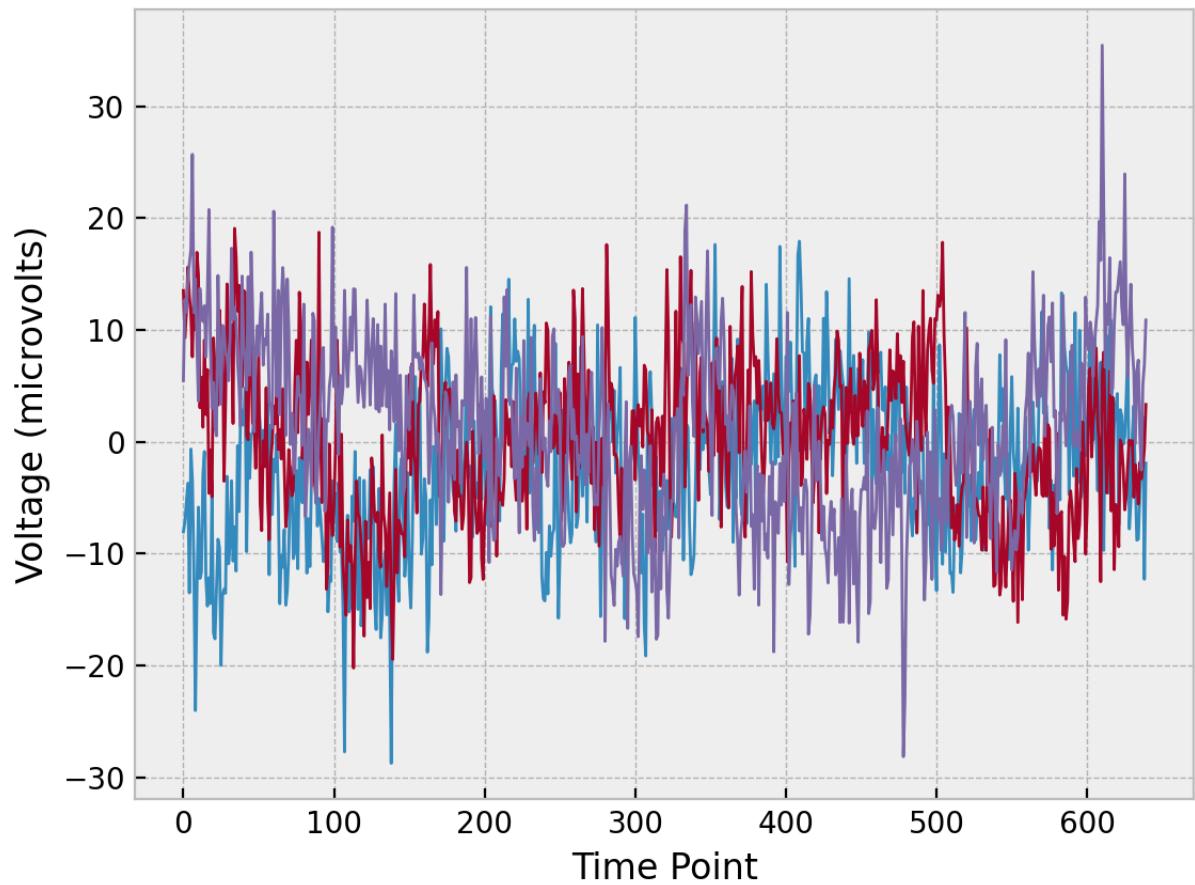
for loops are good for this sort of thing...

```
In [31]: for trial in range(3):
    plt.plot(data[:,trial], label=f"trial {trial}")
plt.xlabel('Time Point')
plt.ylabel('Voltage (microvolts)')
plt.legend();
```

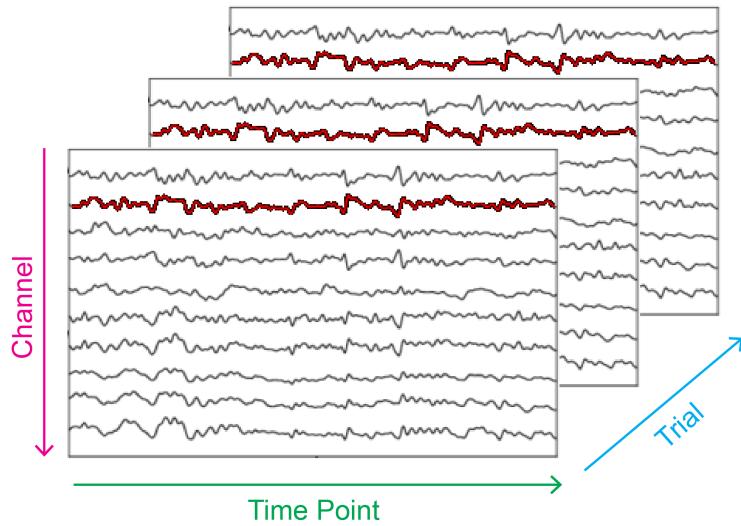


Matplotlib will treat each column of a matrix as a separate plot.

```
In [32]: plt.plot(data)
plt.xlabel('Time Point')
plt.ylabel('Voltage (microvolts)');
```



Let's plot EEGs for channel 1 across all trials overlaid with the trial average.



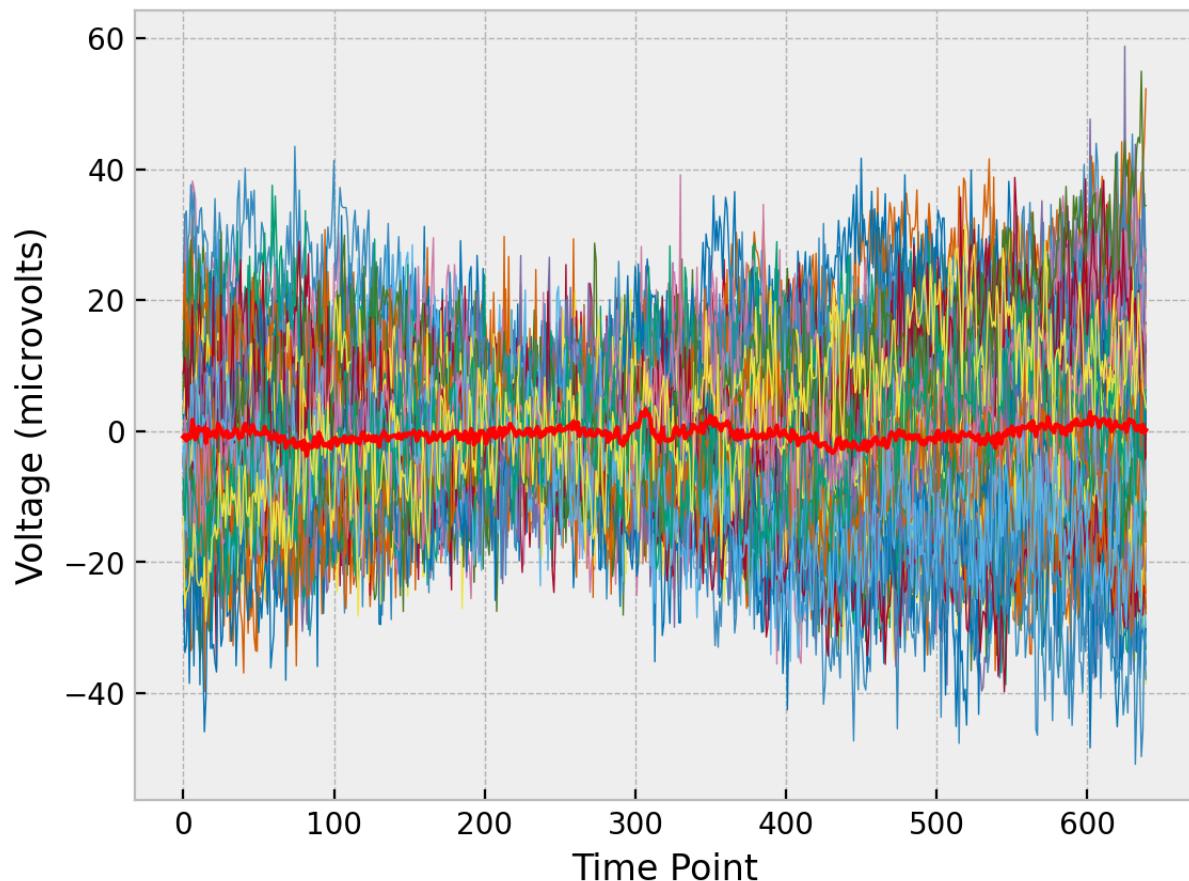
```
In [33]: # data = channel 1, all trials  
data = EEGs[1,:,:]  
data.shape
```

```
Out[33]: (640, 99)
```

```
In [34]: # avg = channel 1 trial average  
avg = data.mean(axis=1)  
avg.shape
```

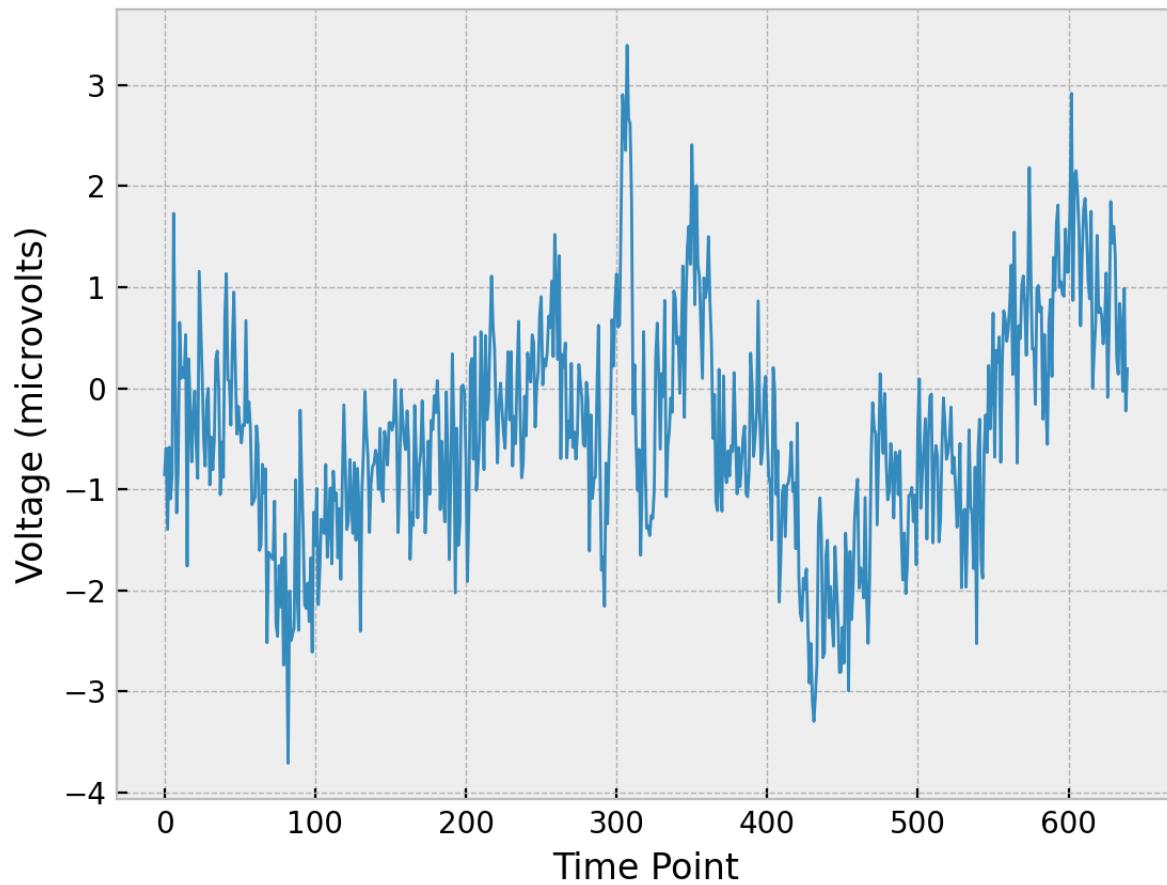
```
Out[34]: (640,)
```

```
In [35]: plt.plot(data, lw=0.5)  
plt.plot(avg, lw=1.5, c='red')  
plt.xlabel('Time Point')  
plt.ylabel('Voltage (microvolts)');
```



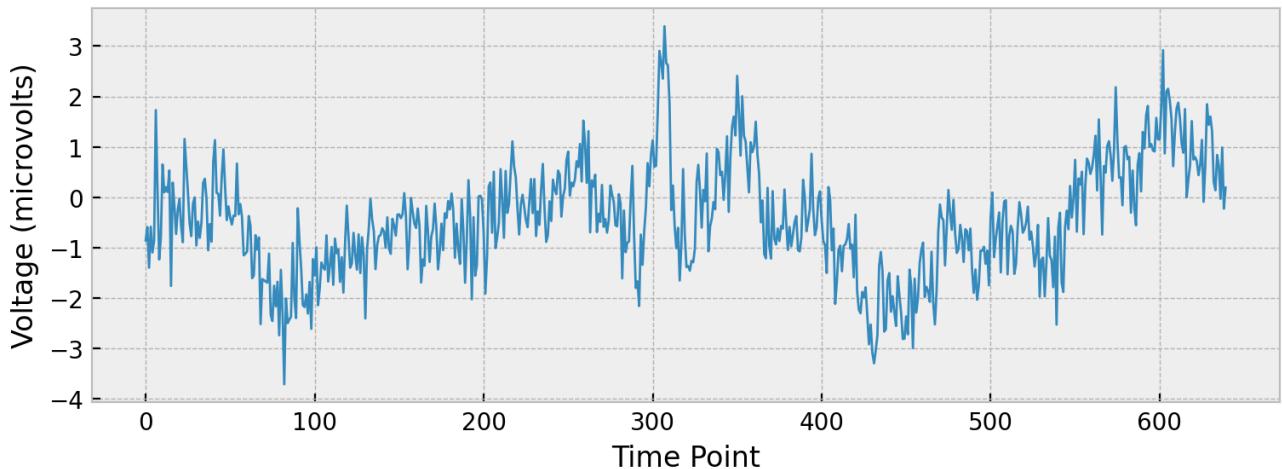
Let's see the trial average alone.

```
In [36]: plt.plot(avg)  
plt.xlabel('Time Point')  
plt.ylabel('Voltage (microvolts)');
```



Let's change the figure size.

```
In [37]: plt.figure(figsize=[9,3])
plt.plot(avg)
plt.xlabel('Time Point')
plt.ylabel('Voltage (microvolts)');
```



Exercise

Get the average EEG across trials for each channel.

```
In [38]: # # EEGs[channel, time, trial]
# EEGs = np.load("data/EEGs.npy")

# avgss = EEGs.mean(axis=???)  
# avgss.shape
```

Exercise Key

Get the average EEG across trials for each channel.

```
In [39]: # EEGs[channel, time, trial]
EEGs = np.load("data/EEGs.npy")

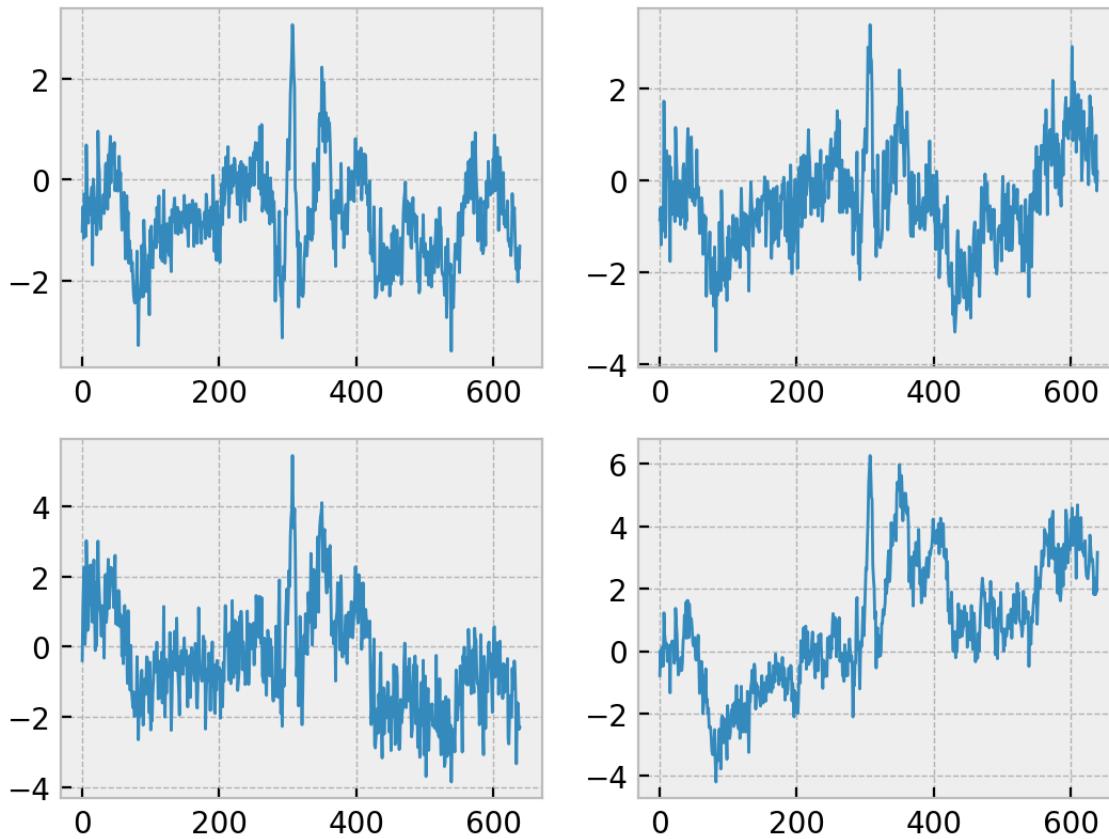
avgss = EEGs.mean(axis=2)

# 64 channels x 640 time pts
avgss.shape
```

```
Out[39]: (64, 640)
```

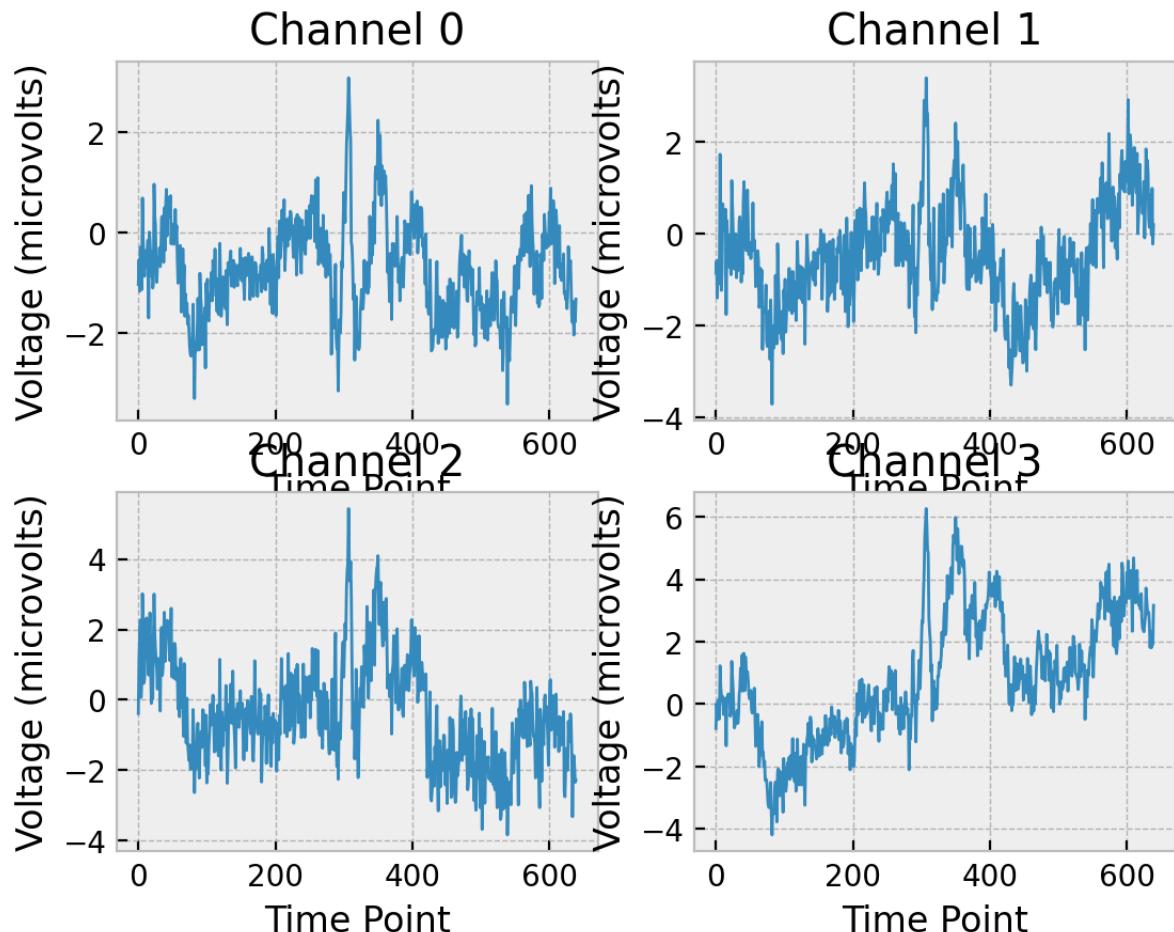
Let's plot the trial averages for channels 0, 1, 2, and 3 in a 2x2 grid.

```
In [40]: plt.subplot(2, 2, 1) # !!! 1-based indexing here
plt.plot(avgss[0,:]) # Last `:` slice is assumed if omitted
plt.subplot(2, 2, 2)
plt.plot(avgss[1])
plt.subplot(2, 2, 3)
plt.plot(avgss[2])
plt.subplot(2, 2, 4)
plt.plot(avgss[3]);
```



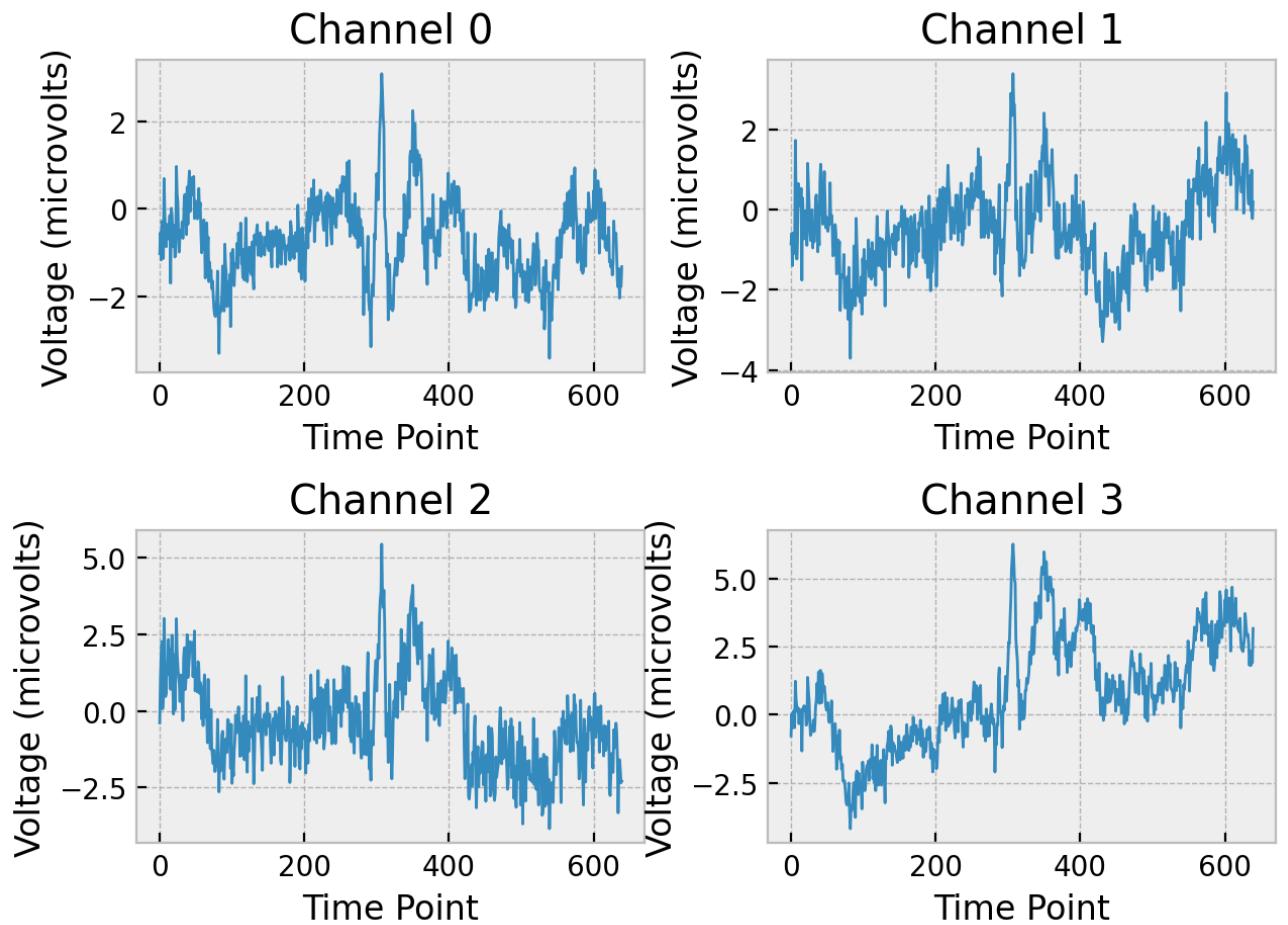
for loops are good for this sort of thing...

```
In [41]: for i in range(4):
    plt.subplot(2, 2, i+1) # i+1 for 1-based index
    plt.plot(avgs[i])
    plt.xlabel('Time Point')
    plt.ylabel('Voltage (microvolts)')
    plt.title(f'Channel {i}')
```



use `plt.tight_layout()` to ensure labels do not overlap.

```
In [42]: for i in range(4):
    plt.subplot(2, 2, i+1)
    plt.plot(avgs[i])
    plt.xlabel('Time Point')
    plt.ylabel('Voltage (microvolts)')
    plt.title(f'Channel {i}')
plt.tight_layout();
```



A Quick Start guide for histograms in Matplotlib

Learning goals

- You will be able to plot 1-D data as a histogram.
- You will be able to style your histograms.
- You will be able to get the histogram bin counts and bin edges as numerical arrays.

Let's consider the weight (g) of mice in both a control and test group (35 mice per group).

```
In [43]: # See np.loadtxt & np.savetxt
# Filepath is relative to this notebook
control_weights, test_weights = np.loadtxt('data/mouse_weights.txt')

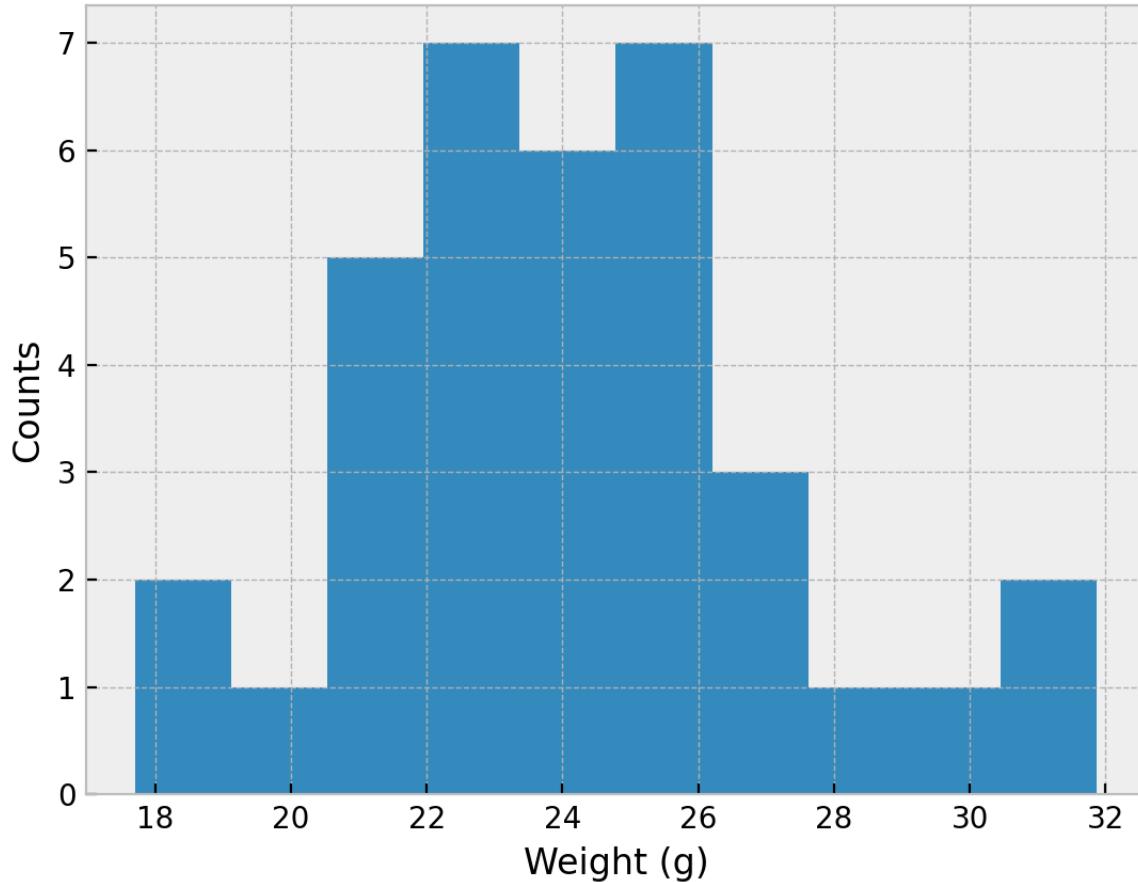
control_weights.shape, test_weights.shape
```

Out[43]: ((35,), (35,))

How might you plot the weights for the control group?

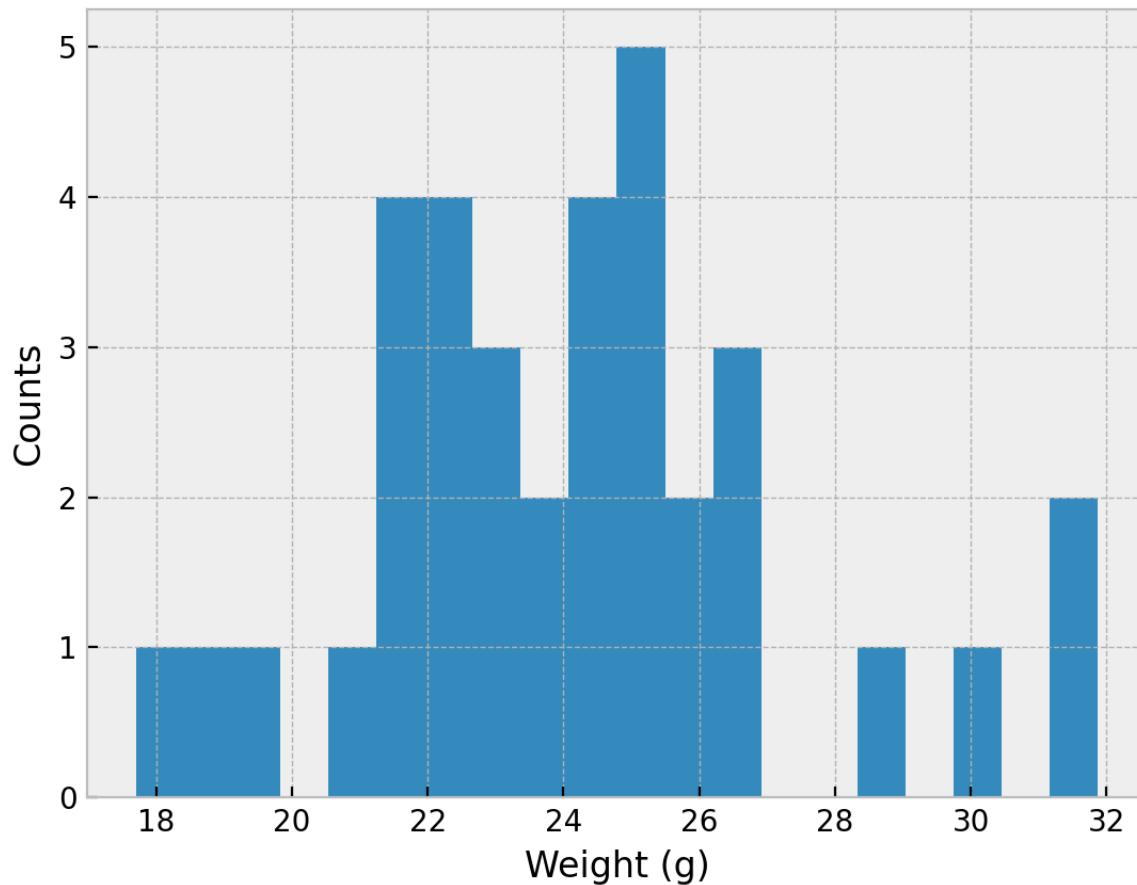
You will often represent one-dimensional data as a **histogram**.

```
In [44]: plt.hist(control_weights)
plt.xlabel('Weight (g)')
plt.ylabel('Counts');
```



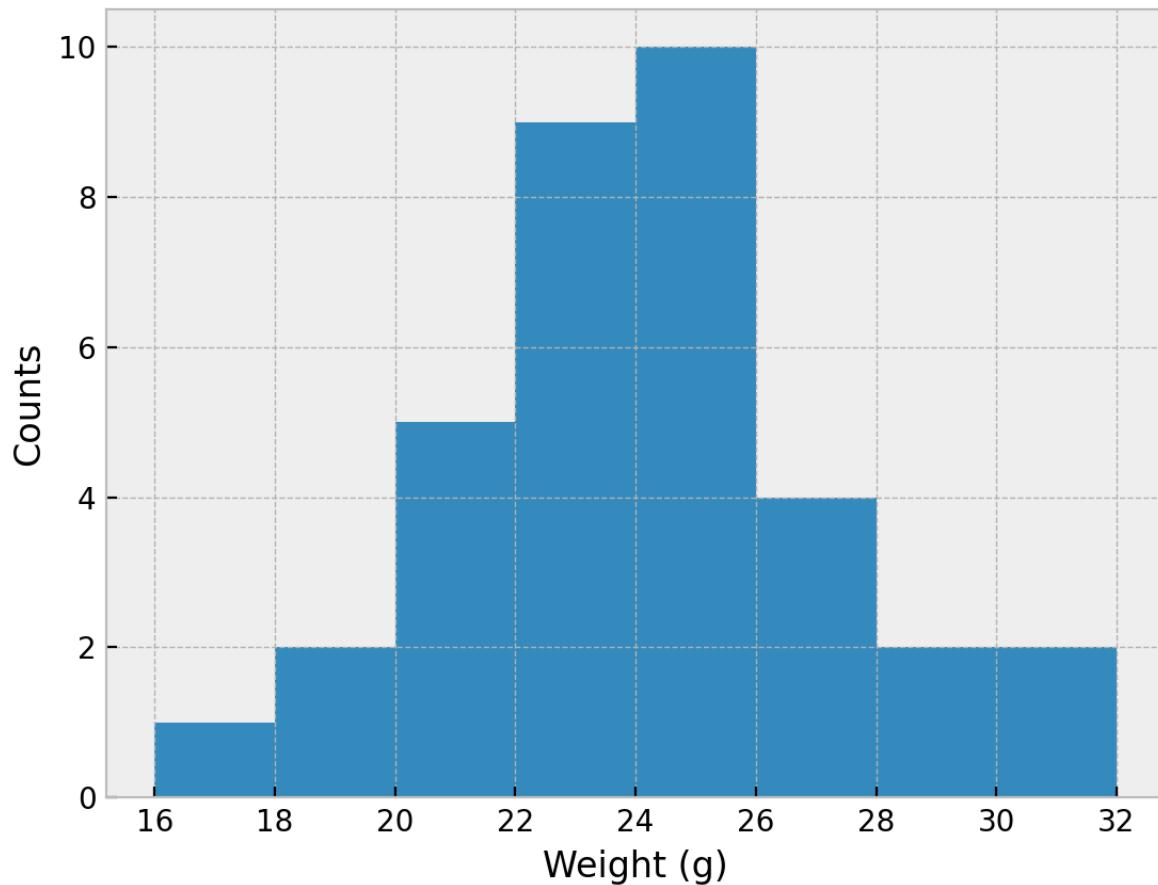
Specify the number of bins (20 in this case).

```
In [45]: plt.hist(control_weights, 20)
plt.xlabel('Weight (g)')
plt.ylabel('Counts');
```



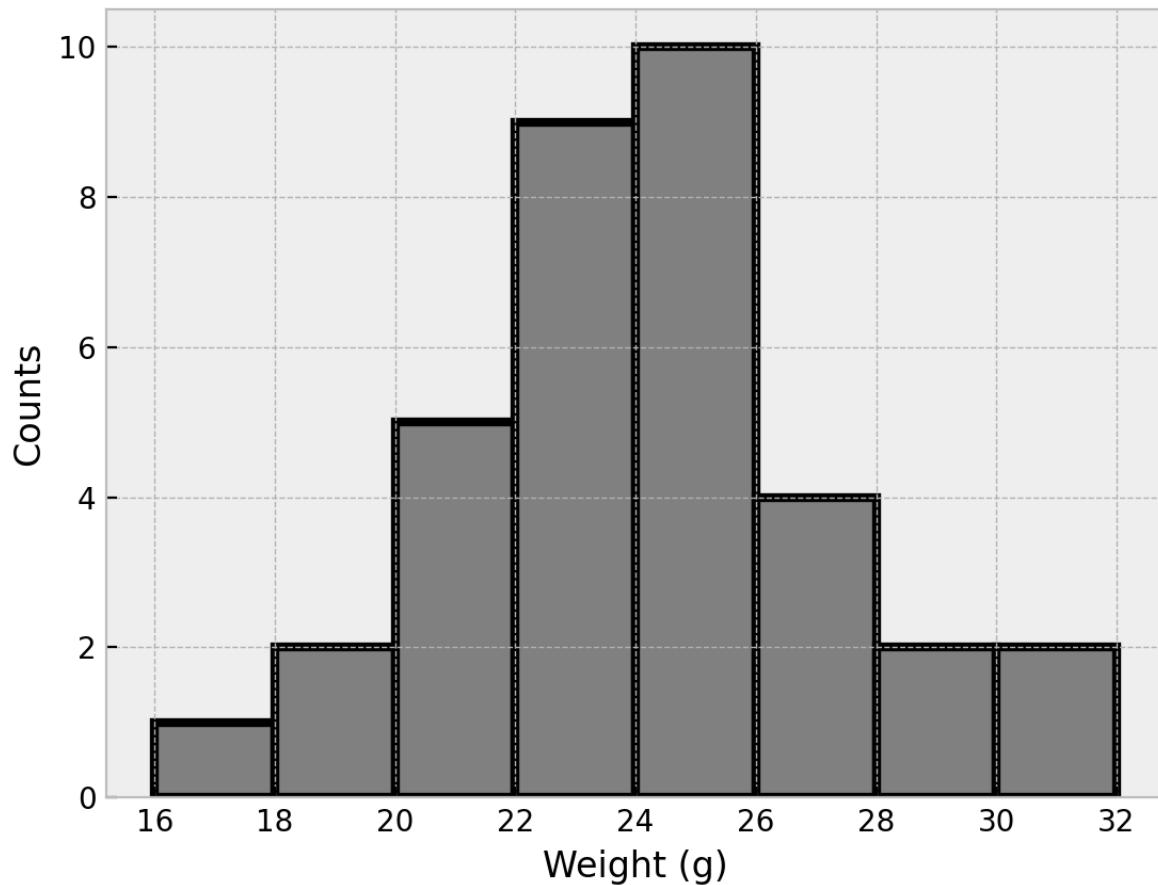
Specify the bin edges from 16-32 in steps of 2.

```
In [46]: bin_edges = np.arange(16, 33, 2)
plt.hist(control_weights, bin_edges)
plt.xlabel('Weight (g)')
plt.ylabel('Counts');
```



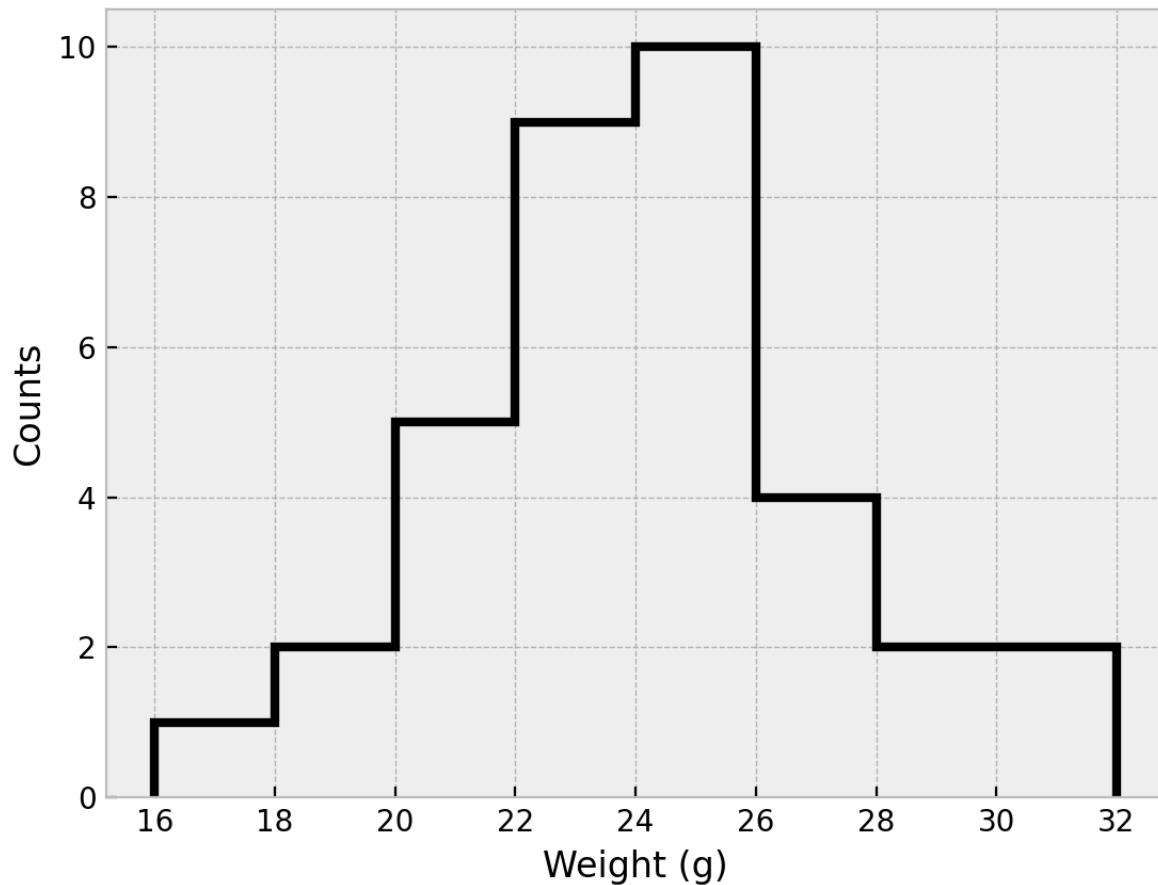
Specify face color using `fc=`, edge color using `ec=`, edge line width using `lw=`

```
In [47]: bin_edges = np.arange(16, 33, 2)
plt.hist(control_weights, bin_edges, fc='gray', ec='black', lw=3)
plt.xlabel('Weight (g)')
plt.ylabel('Counts');
```



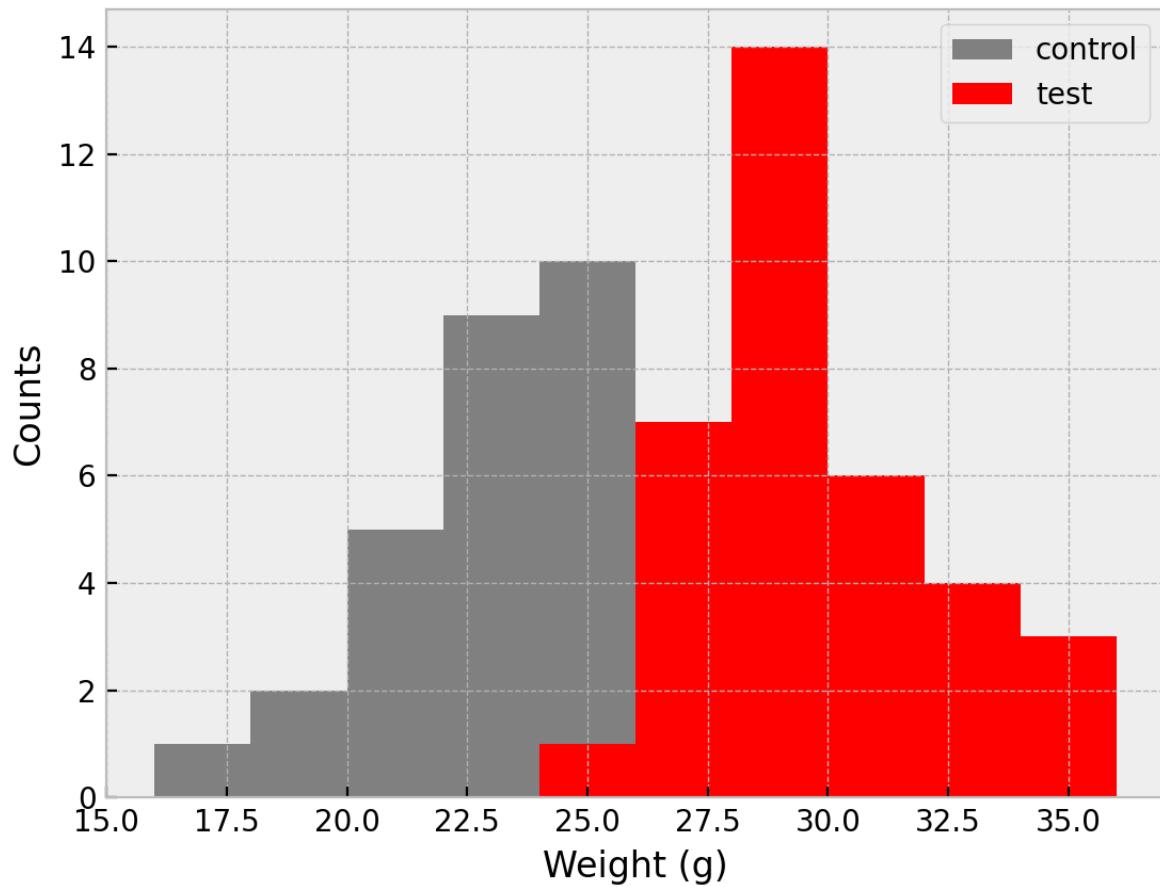
Specify histogram style with `histtype=`

```
In [48]: bin_edges = np.arange(16, 33, 2)
plt.hist(control_weights, bin_edges, ec='black', lw=3, histtype='step')
plt.xlabel('Weight (g)')
plt.ylabel('Counts');
```



Plot histograms for both the control and test groups on the same axes.

```
In [49]: bin_edges = np.arange(16, 38, 2)
plt.hist(control_weights, bin_edges, fc='gray', label='control')
plt.hist(test_weights, bin_edges, fc='red', label='test')
plt.xlabel('Weight (g)')
plt.ylabel('Counts')
plt.legend();
```



Specify opacity using `alpha=`

```
In [50]: bin_edges = np.arange(16, 38, 2)
plt.hist(control_weights, bin_edges, fc='gray', alpha=0.5, label='control')
plt.hist(test_weights, bin_edges, fc='red', alpha=0.5, label='test')
plt.xlabel('Weight (g)')
plt.ylabel('Counts')
plt.legend();
```

