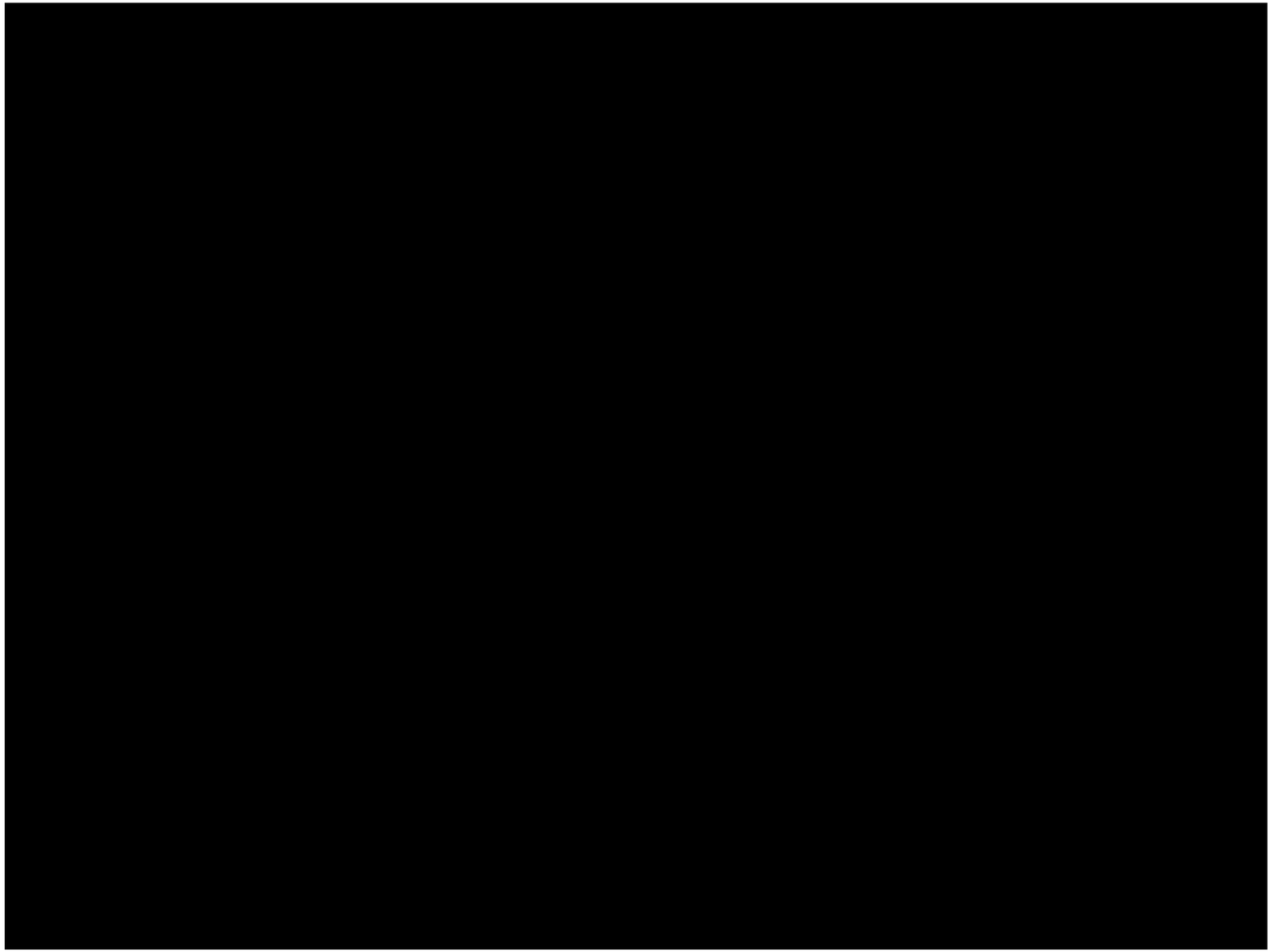
Making figures with Python



Dudas et al. 2017, Nature

Spring 2022
PCfB Class 10
March 25, 2022

Outline

- Benefits of Python for figures
- Intro to Matplotlib module

Why use Python?

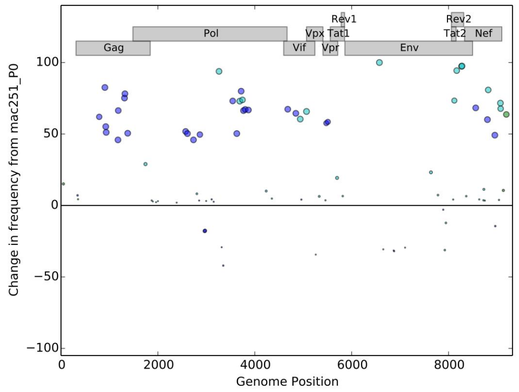
- Highly customizable
- Automated, easy to rerun
- Integrate into analysis
- Open science compatible

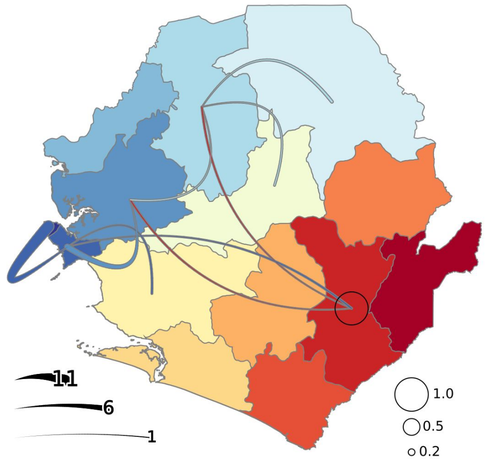
Using matplotlib

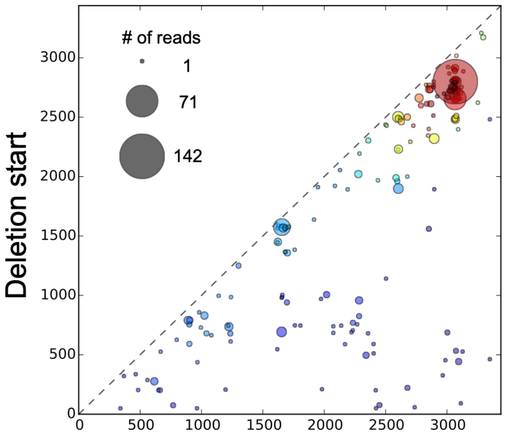
matplotlib

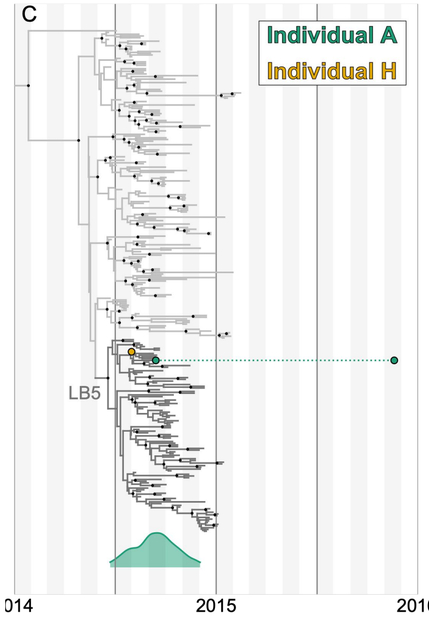
- Powerful plotting module

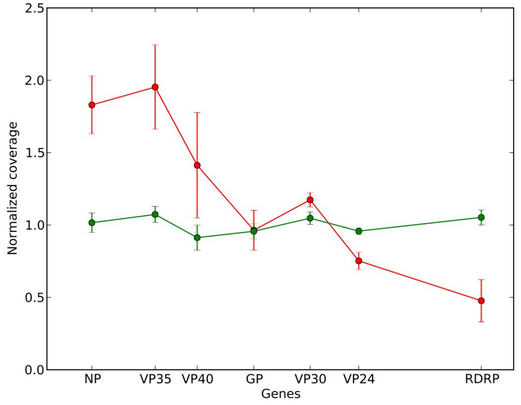
- Preloaded in Anaconda installations

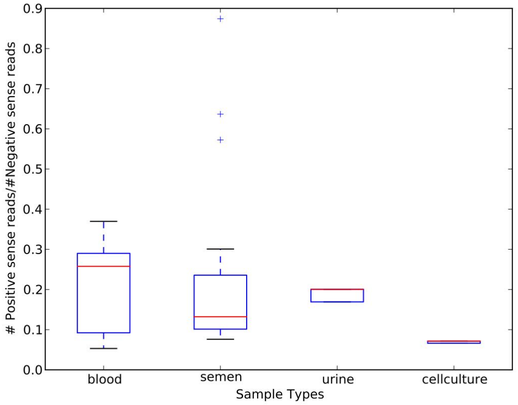












- `import matplotlib.pyplot as plt`

- Recommended ways to use Matplotlib:
  - Stand-alone scripts
  - Jupyter notebook

Plot Objects

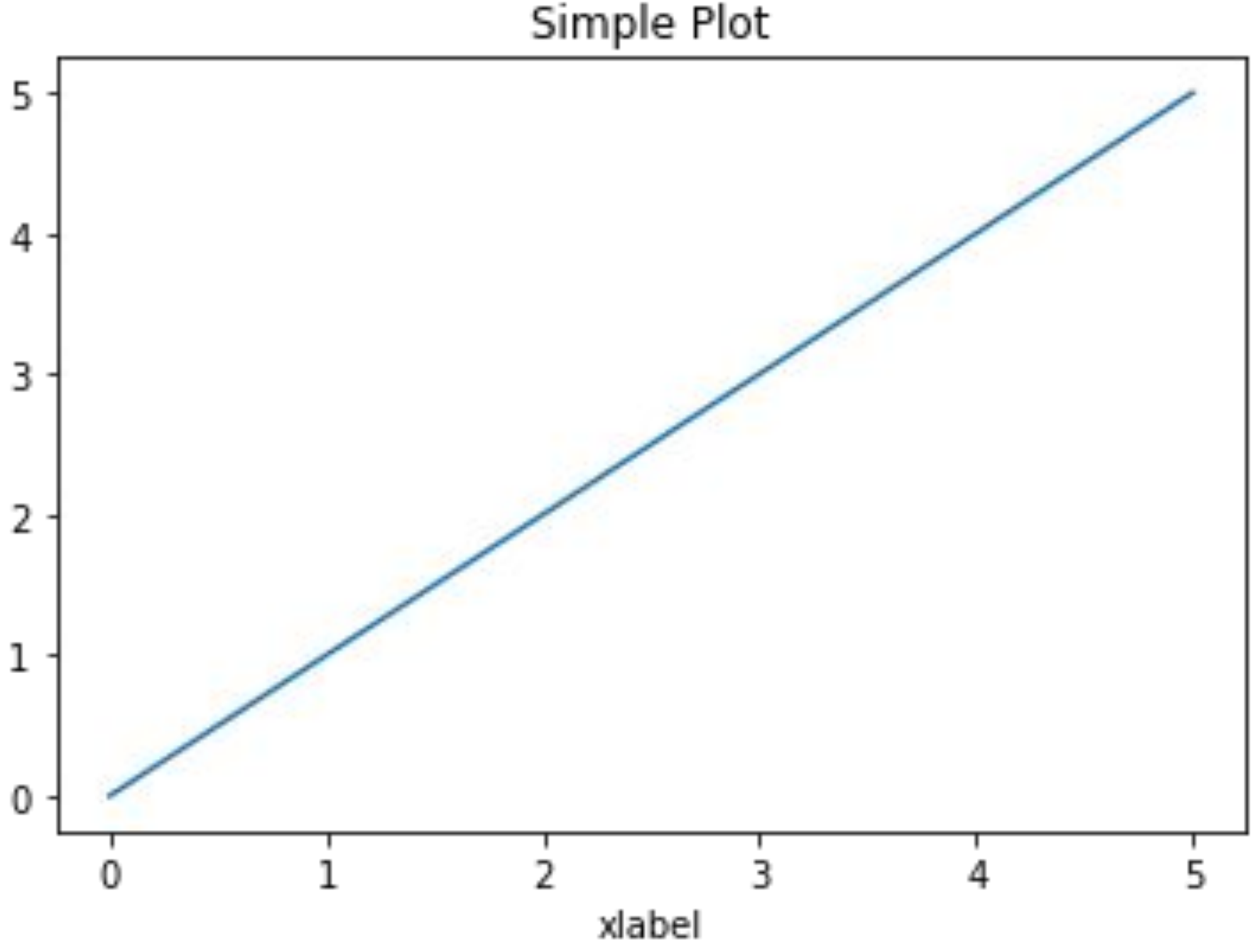
- **`Figure`** objects
  - Top-level container for plot elements
  - Can contain multiple Axes objects (i.e. plots)
- **`Axes`** objects
  - One per graph/plot
  - Contains most figure elements

State-machine interface

```
import matplotlib.pyplot as plt
plt.plot([0,5], [0,5])
plt.xlabel('xlabel')
plt.title('Simple Plot')
```

Object-oriented approach

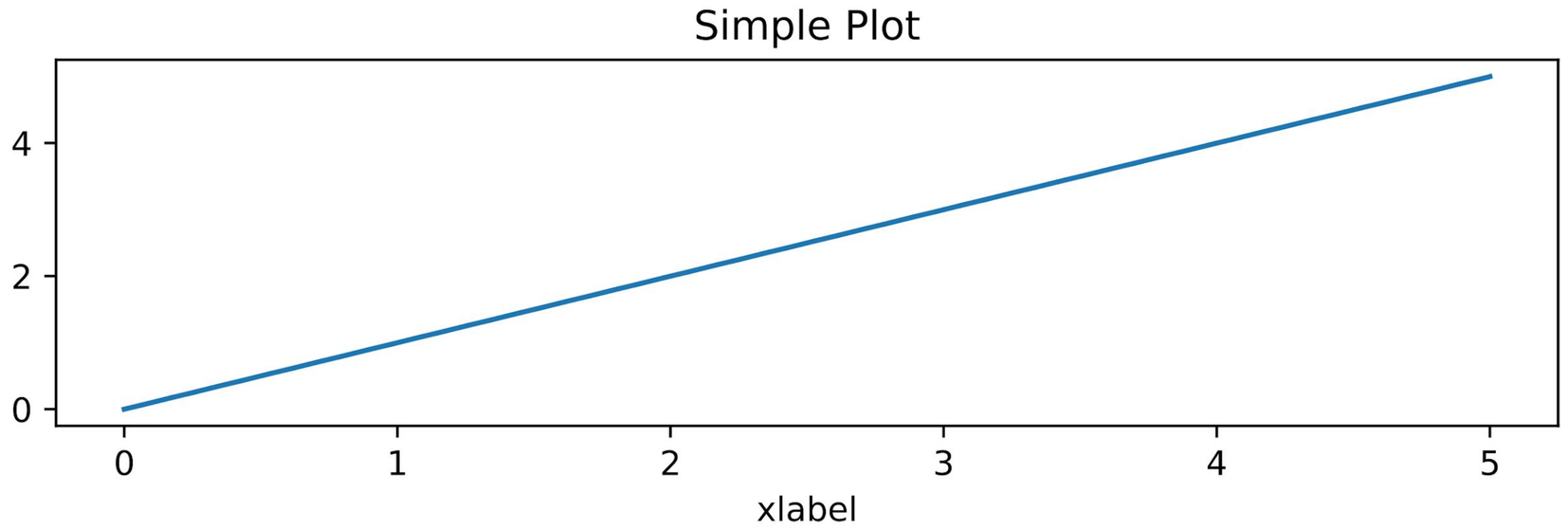
```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.plot([0,5], [0,5])
ax.set_xlabel('xlabel')
ax.set_title('Simple Plot')
```


Simple Plot
5
4
3
2
1
0
0
1
2
3
4
5
xlabel

Options at initialization

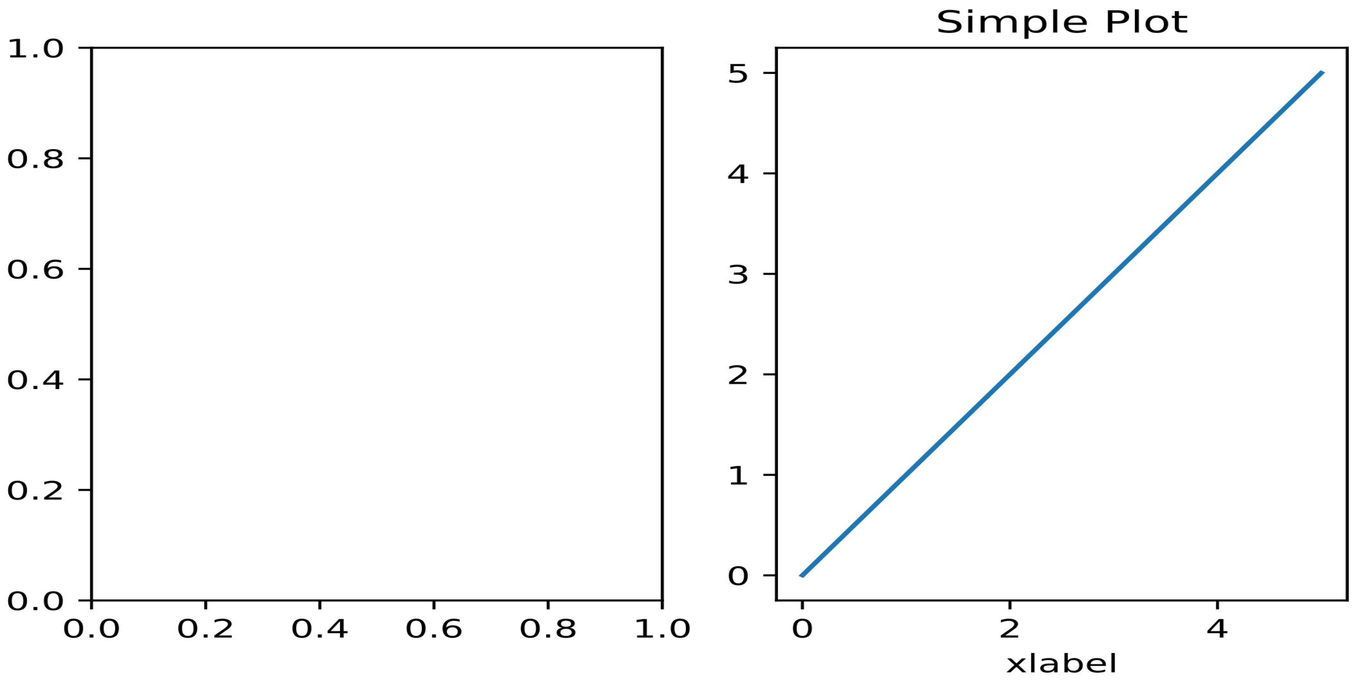
```
fig, ax = plt.subplots()

fig, ax = plt.subplots(figsize=(8,2))

fig, ax = plt.subplots(1,2)

fig, ax = plt.subplots(1,2, figsize=(8,2))
```

Control figure size



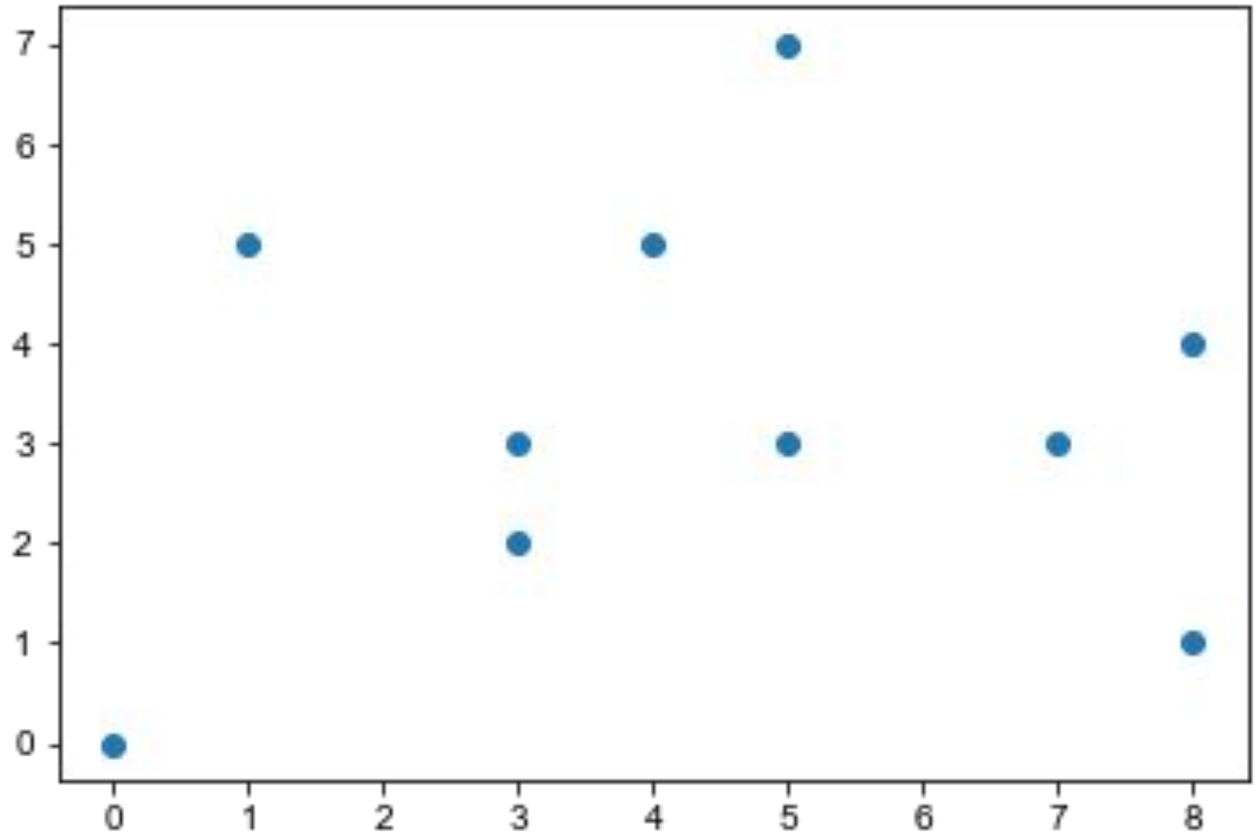

```
fig, ax = plt.subplots(figsize=(8,2))

ax.plot([0,5], [0,5])

ax.set_xlabel('xlabel')

ax.set_title('Simple Plot')
```

Multiple axes per figure



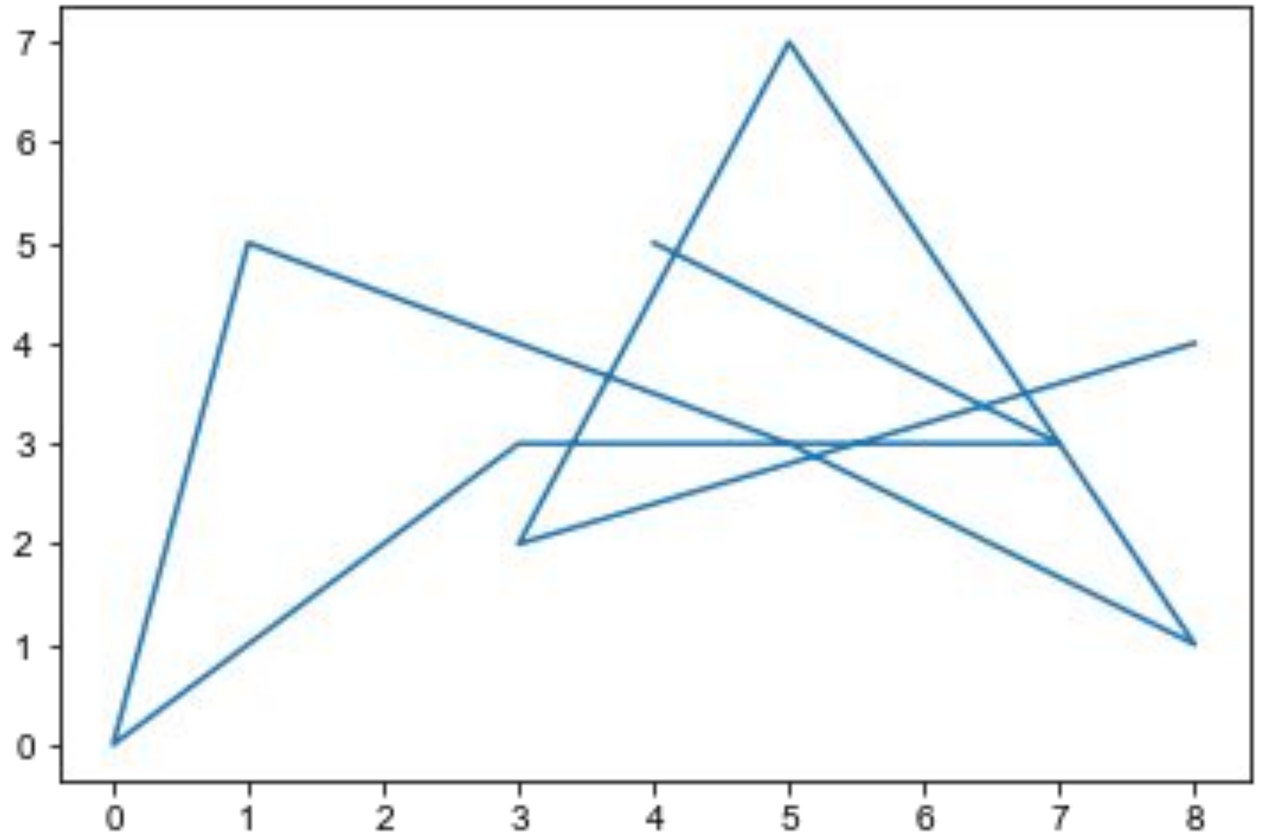

```
fig, ax = plt.subplots(1,2)

ax[1].plot([0,5], [0,5])

ax[1].set_xlabel('xlabel')

ax[1].set_title('Simple Plot')
```

Scatterplot



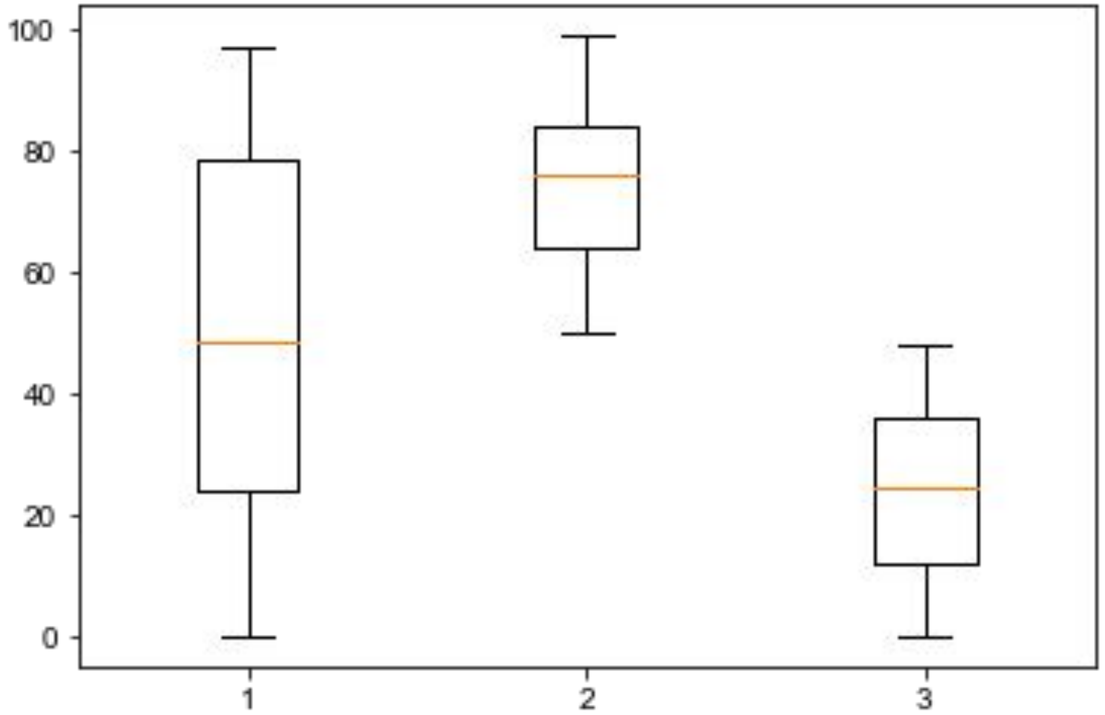

```
x = random.choices(range(10), k=10)
y = random.choices(range(10), k=10)
ax.scatter(x, y)
```

Lineplot



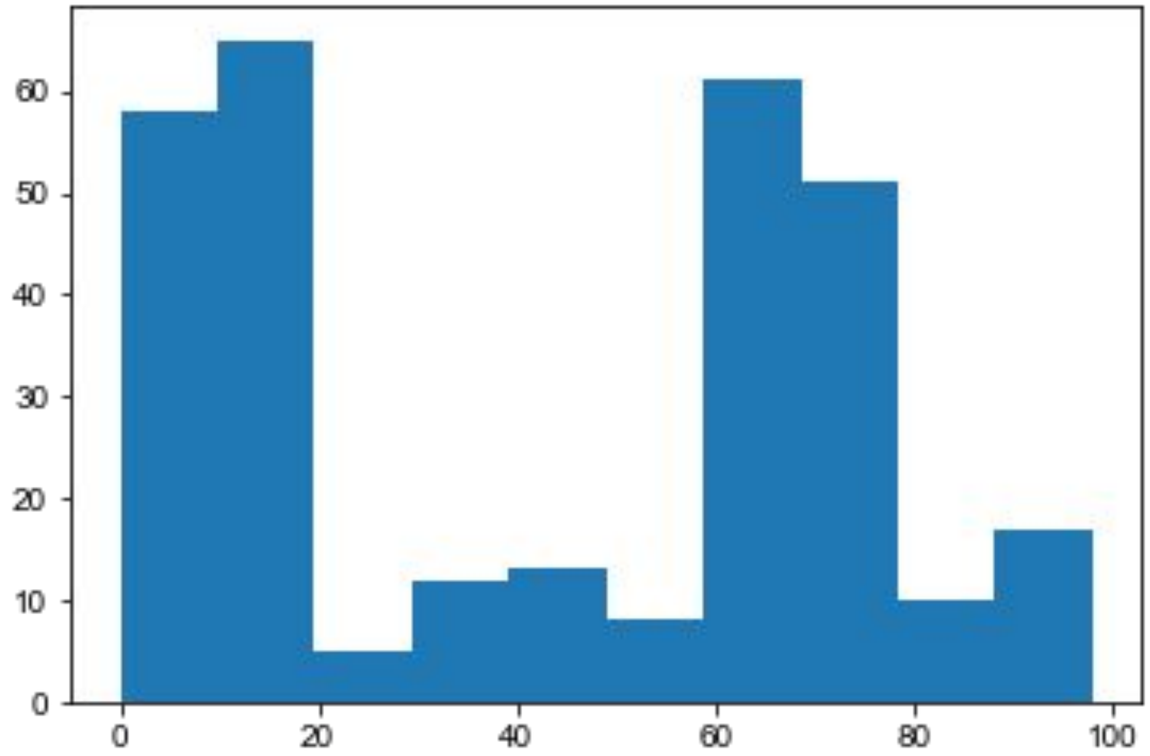

```
x = random.choices(range(10), k=10)
y = random.choices(range(10), k=10)
ax.plot(x, y)
```

Boxplot



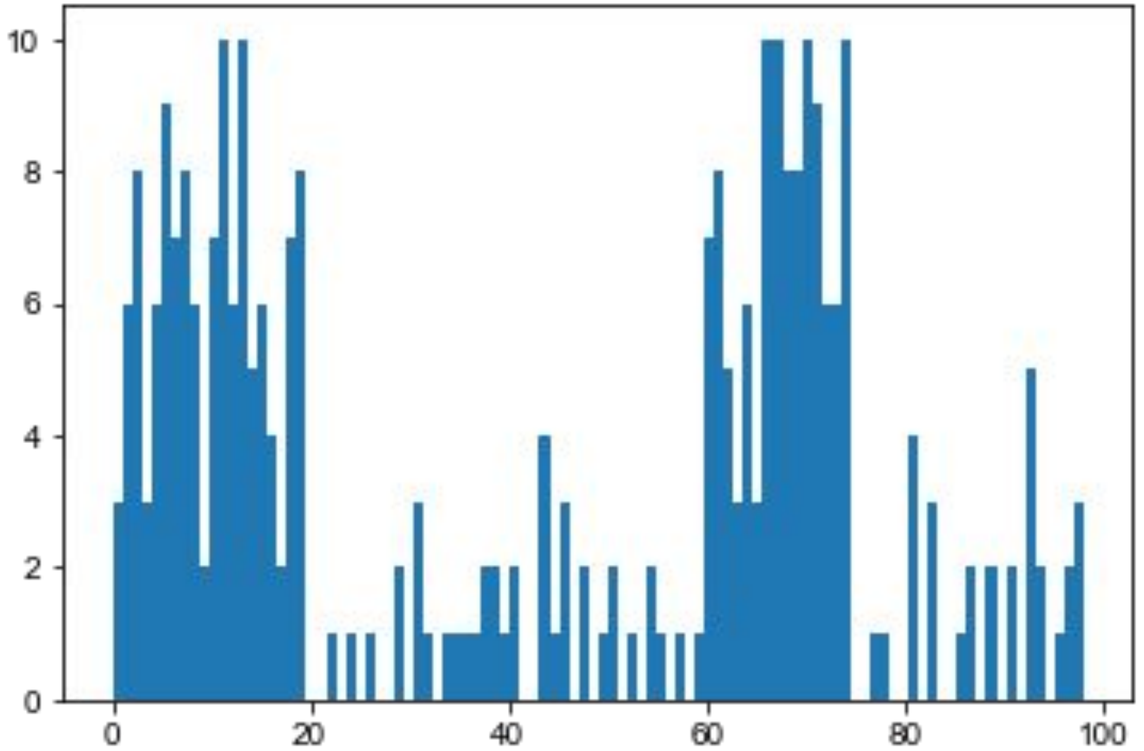

```
d = [random.choices(range(100), k=100),
     random.choices(range(50,100), k=100),
     random.choices(range(50), k=100)]
ax.boxplot(d)
```

Histogram



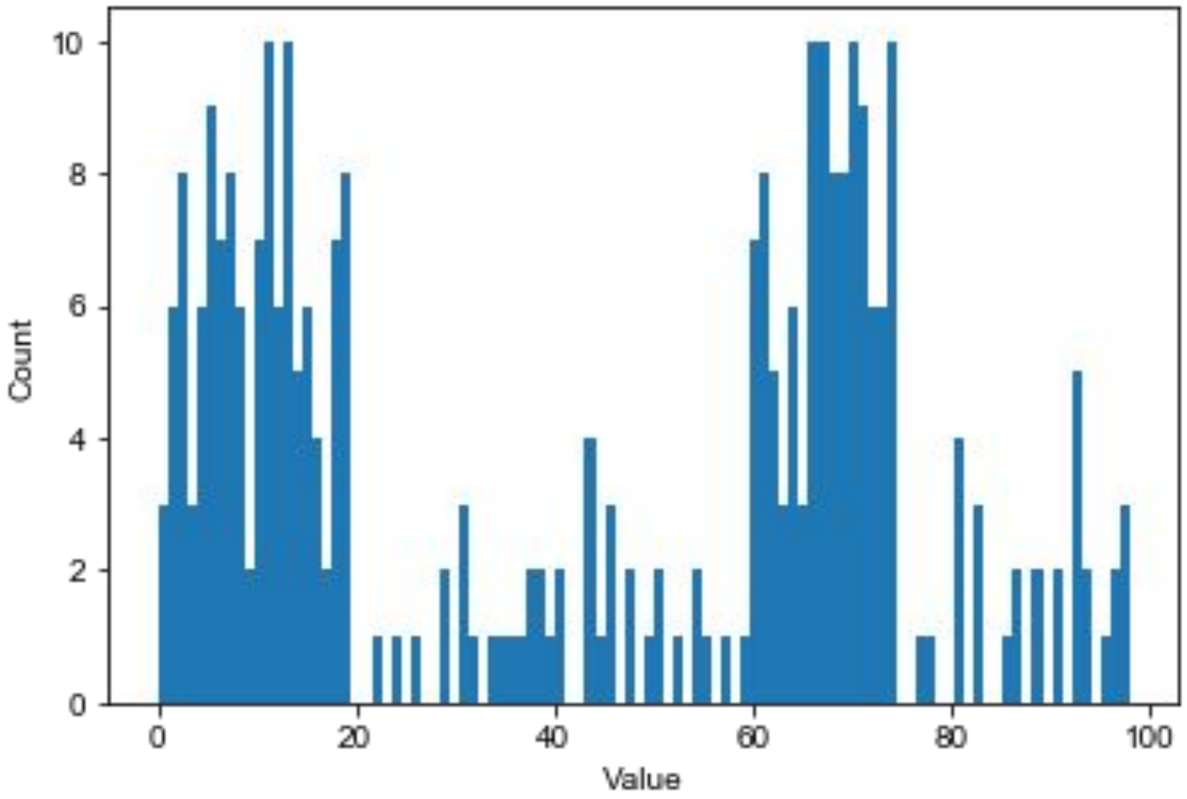

```
o = random.choices(range(20), k=100) +
    random.choices(range(100), k=100) +
    random.choices(range(60, 75), k=100)
ax.hist(o)
```

Histogram



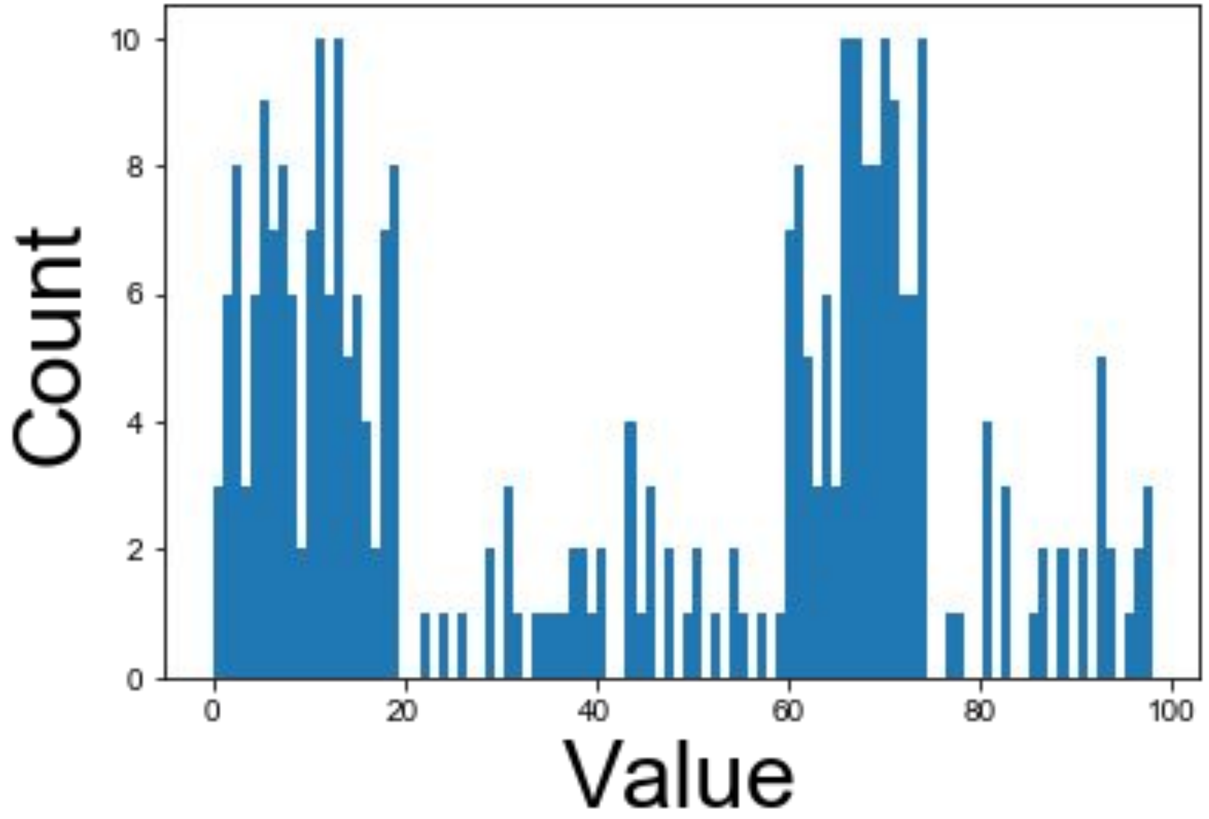

```
o = random.choices(range(20), k=100) +
    random.choices(range(100), k=100) +
    random.choices(range(60, 75), k=100)
ax.hist(o, bins=100)
```

Adding axis labels



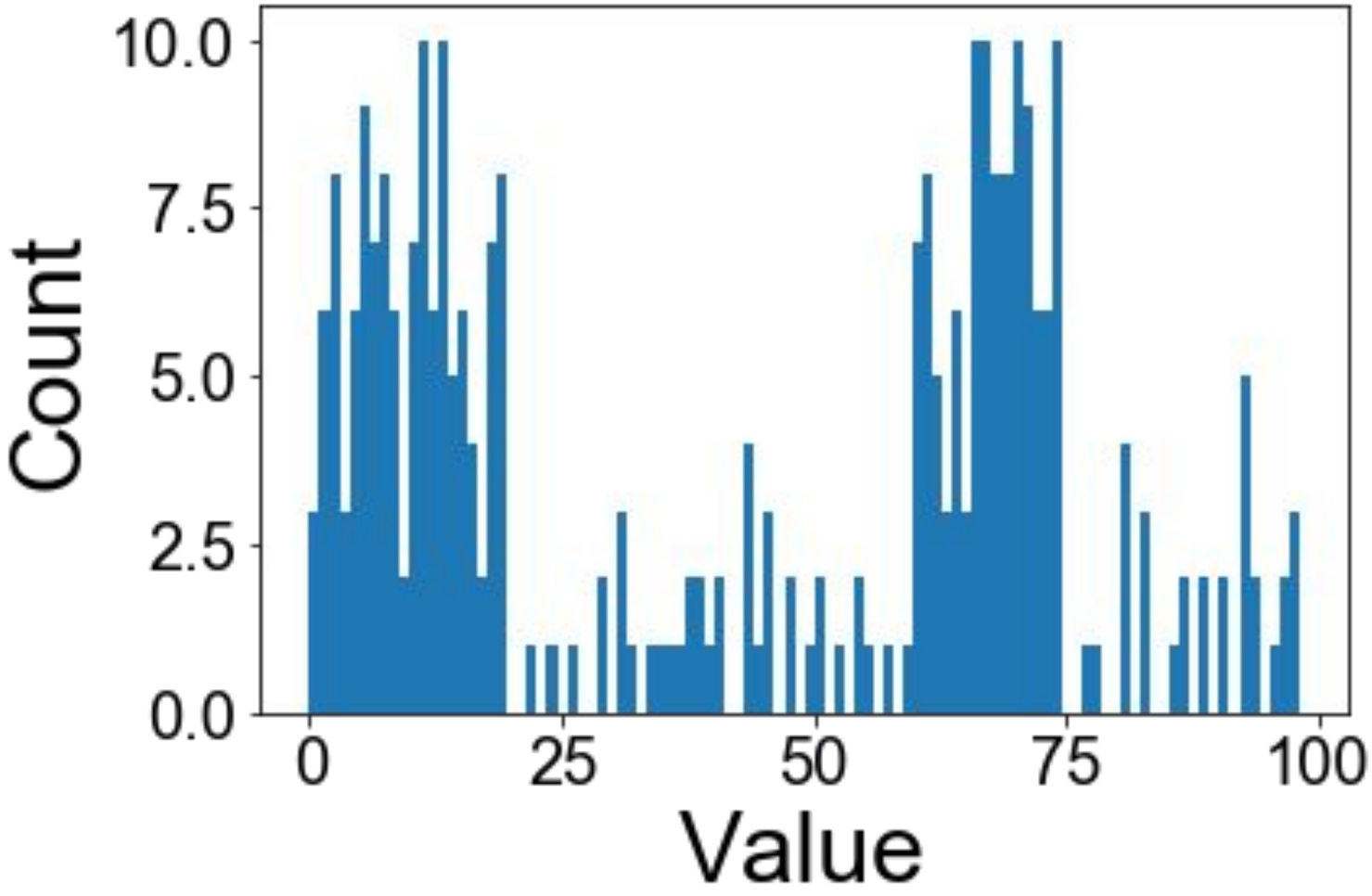

```
ax.set_xlabel("Value")
ax.set_ylabel("Count")
```

Adding axis labels



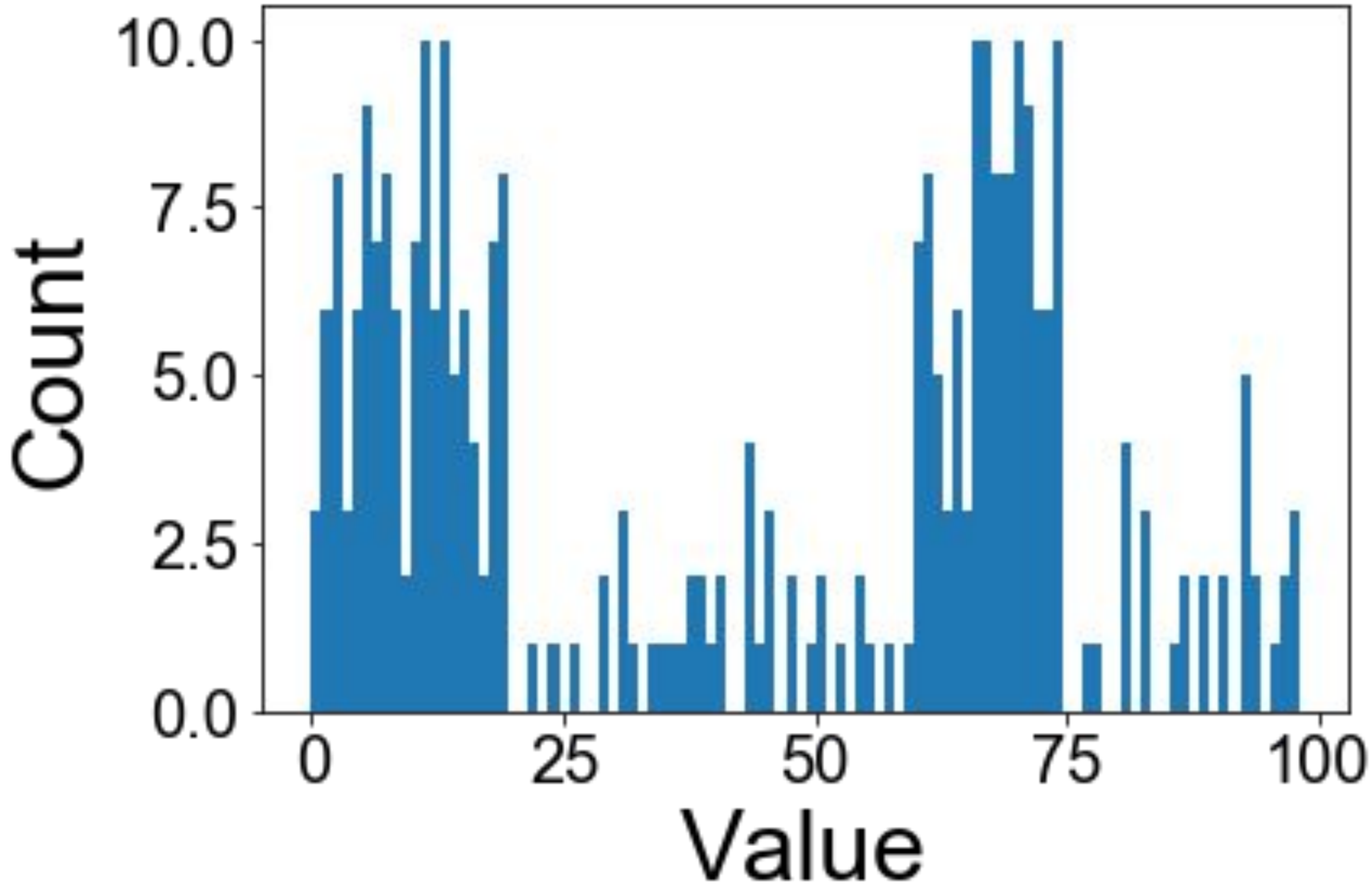

```
ax.set_xlabel("Value", fontsize=30)
ax.set_ylabel("Count", fontsize=30)
```

Adding axis labels




```
ax.tick_params(labelsize=20)
```

Adding axis labels




```
ax.tick_params(axis='both', which='major', labelsize=20)
```

Axes object methods

['acorr', 'add_artist', 'add_callback', 'add_collection', 'add_container', 'add_line', 'add_patch', 'add_table', 'aname', 'annotate', 'apply_aspect', 'arrow', 'artists', 'autoscale', 'autoscale_view', 'axes', 'axesPatch', 'axhline', 'axhspan', 'axis', 'axison', 'axvline', 'axvspan', 'bar', 'barbs', 'barh', 'bbox', 'boxplot', 'broken_barh', 'callbacks', 'can_pan', 'can_zoom', 'change_geometry', 'cla', 'clabel', 'clear', 'clipbox', 'cohere', 'colNum', 'collections', 'containers', 'contains', 'contains_point', 'contour', 'contourf', 'convert_xunits', 'convert_yunits', 'csd', 'dataLim', 'drag_pan', 'draw', 'draw_artist', 'end_pan', 'errorbar', 'eventplot', 'eventson', 'figbox', 'figure', 'fill', 'fill_between', 'fill_betweenx', 'findobj', 'fmt_xdata', 'fmt_ydata', 'format_coord', 'format_xdata', 'format_ydata', 'get_adjustable', 'get_agg_filter', 'get_alpha', 'get_anchor', 'get_animated', 'get_aspect', 'get_autoscale_on', 'get_autoscalex_on', 'get_autoscaley_on', 'get_axes', 'get_axes_locator', 'get_axis_bgcolor', 'get_axisbelow', 'get_children', 'get_clip_box', 'get_clip_on', 'get_clip_path', 'get_contains', 'get_cursor_props', 'get_data_ratio', 'get_data_ratio_log', 'get_default_bbox_extra_artists', 'get_figure', 'get_frame_on', 'get_geometry', 'get_gid', 'get_images', 'get_label', 'get_legend', 'get_legend_handles_labels', 'get_lines', 'get_navigate', 'get_navigate_mode', 'get_path_effects', 'get_picker', 'get_position', 'get_rasterization_zorder', 'get_rasterized', 'get_renderer_cache', 'get_shared_x_axes', 'get_shared_y_axes', 'get_sketch_params', 'get_snap', 'get_subplotspec', 'get_tightbbox', 'get_title', 'get_transform', 'get_transformed_clip_path_and_affine', 'get_url', 'get_visible', 'get_window_extent', 'get_xaxis', 'get_xaxis_text1_transform', 'get_xaxis_text2_transform', 'get_xaxis_transform', 'get_xbound', 'get_xgridlines', 'get_xlabel', 'get_xlim', 'get_xmajorticklabels', 'get_xminorticklabels', 'get_xscale', 'get_xticklabels', 'get_xticklines', 'get_xticks', 'get_yaxis', 'get_yaxis_text1_transform', 'get_yaxis_text2_transform', 'get_yaxis_transform', 'get_ybound', 'get_ygridlines', 'get_ylabel', 'get_ylim', 'get_ymajorticklabels', 'get_yminorticklabels', 'get_yscale', 'get_yticklabels', 'get_yticklines', 'get_yticks', 'get_zorder', 'grid', 'has_data', 'have_units', 'hexbin', 'hist', 'hist2d', 'hitlist', 'hlines', 'hold', 'ignore_existing_data_limits', 'images', 'imshow', 'in_axes', 'invert_xaxis', 'invert_yaxis', 'is_figure_set', 'is_first_col', 'is_first_row', 'is_last_col', 'is_last_row', 'is_transform_set', 'ishold', 'label_outer', 'legend', 'legend_', 'lines', 'locator_params', 'loglog', 'margins', 'matshow', 'minorticks_off', 'minorticks_on', 'name', 'numCols', 'numRows', 'patch', 'patches', 'pchanged', 'pcolor', 'pcolorfast', 'pcolormesh', 'pick', 'pickable', 'pie', 'plot', 'plot_date', 'properties', 'psd', 'quiver', 'quiverkey', 'redraw_in_frame', 'relim', 'remove', 'remove_callback', 'reset_position', 'rowNum', 'scatter', 'semilogx', 'semilogy', 'set', 'set_adjustable', 'set_agg_filter', 'set_alpha', 'set_anchor', 'set_animated', 'set_aspect', 'set_autoscale_on', 'set_autoscalex_on', 'set_autoscaley_on', 'set_axes', 'set_axes_locator', 'set_axis_bgcolor', 'set_axis_off', 'set_axis_on', 'set_axisbelow', 'set_clip_box', 'set_clip_on', 'set_clip_path', 'set_color_cycle', 'set_contains', 'set_cursor_props', 'set_figure', 'set_frame_on', 'set_gid', 'set_label', 'set_lod', 'set_navigate', 'set_navigate_mode', 'set_path_effects', 'set_picker', 'set_position', 'set_rasterization_zorder', 'set_rasterized', 'set_sketch_params', 'set_snap', 'set_subplotspec', 'set_title', 'set_transform', 'set_url', 'set_visible', 'set_xbound', 'set_xlabel', 'set_xlim', 'set_xmargin', 'set_xscale', 'set_xticklabels', 'set_xticks', 'set_ybound', 'set_ylabel', 'set_ylim', 'set_ymargin', 'set_yscale', 'set_yticklabels', 'set_yticks', 'set_zorder', 'specgram', 'spines', 'spy', 'stackplot', 'start_pan', 'stem', 'step', 'streamplot', 'table', 'tables', 'text', 'texts', 'tick_params', 'ticklabel_format', 'title', 'titleOffsetTrans', 'transAxes', 'transData', 'transLimits', 'transScale', 'tricontour', 'tricontourf', 'tripcolor', 'triplot', 'twinx', 'twiny', 'update', 'update_datalim', 'update_datalim_bounds', 'update_datalim_numerix', 'update_from', 'update_params', 'viewLim', 'vlines', 'xaxis', 'xaxis_date', 'xaxis_inverted', 'xcorr', 'yaxis', 'yaxis_date', 'yaxis_inverted', 'zorder']

Viewing your plots

Plots occur inline with code in jupyter notebook

In [2]:
```
1 import matplotlib.pyplot as plt
2 plt.get_backend()
```

Out[2]: 'module://ipykernel.pylab.backend_inline'

```
%matplotlib inline
```

Saving your plots

```
fig.savefig('name.pdf')
```

Saving your plots

```
fig.savefig('name.pdf',
            bbox_inches='tight')

fig.savefig('name.png',
bbox_inches='tight', dpi=200)
```