Table of contents

```
import matplotlib as mpl
import matplotlib.font_manager as font_manager
from IPython.core.display import HTML
import matplotlib.pyplot as plt
import numpy as np
from directory_tree import display_tree
```

```
plt.rcParams.update({'font.size': 12,
                     'lines.linewidth': 1,
                     'lines.markersize': 10,
                     'axes.labelsize': 11,
                     'xtick.labelsize' : 10,
                     'ytick.labelsize' : 10,
                     'xtick.top' : True,
                     'xtick.direction' : 'in',
                     'ytick.right' : True,
                     'ytick.direction' : 'in',})

%config InlineBackend.figure_format = 'retina'
```

```
def get_size(w,h):
    return((w/2.54,h/2.54))
```

```
# Generate fake experimental data
x = np.linspace(0, 10, 100)
y = np.sin(x) + np.random.normal(0, 0.2, size=100)

# Set up the fancy plot
fig, ax = plt.subplots(figsize=get_size(8, 6),dpi=150)
ax.plot(x, y, color='blue', linewidth=2, label='EXP',alpha=0.5)

# Add some fancy elements to the plot
ax.fill_between(x, y, color='lightgray')
ax.scatter(x, y, color='blue', s=30, label='DATA ',alpha=0.5)

ax.set_xlabel('Time')
ax.set_ylabel('Measurement')

ax.legend(loc='lower right')
```

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

ax.grid(color='gray', linestyle='--', linewidth=0.5)

ax.annotate('Interesting\nPoint', xy=(4.8, 0), xytext=(4.8, 0.7),
            arrowprops=dict(facecolor='black', arrowstyle='->'),
            fontsize=10, ha='center')
plt.savefig("../figure2.pdf",bbox_inches = 'tight')
plt.show()
```