

96% of All the Things You'd Ever Want to Plot in Python

Set up

```
import matplotlib.pyplot as plt
import seaborn as sns

fig, axes = plt.subplots(2, 3, figsize = (13,9))

y1 = [1,2,3,4,5]; x1 = [0]*len(y1)
y2 = [3,3,3,3,4,5,5,5,5,6,7,7,8,9,10,10,10]; x2 = [1]*len(y2)
```

matrix containing each of the plots. Use as ax argument to other functions

rows, columns

width,height

```
# Add text to the plot using LaTeX
axes[0,0].text(-0.3, 8, '..latex..')
```

```
# Remove the top and right borders
sns.despine(ax = axes[0, 0])
```

```
sns.boxplot(x = x1 + x2,
            y = y1 + y2,
            ax = axes[0, 0])
```

```
sns.violinplot(x = x1 + x2,
               y = y1 + y2,
               ax = axes[0, 1])
```

```
axes[0, 1].annotate('Violin plot',
                   xy = (0.8, 10),
                   xytext = (0.2, 12),
                   arrowprops =
                       {'arrowstyle': '->'})
```

Position of arrowhead

position of text

Whether to plot a density plot

```
sns.distplot(y1, hist = False,
             kde = True,
             ax = axes[0, 2])
```

```
sns.distplot(y2, hist = True,
             kde = True, bins = 10,
             norm_hist = True,
             ax = axes[0, 2])
```

Number of bars in the histogram

Do we want to plot a density (sums to 1) or counts

```
# Can also do .semilogy()
axes[0, 2].semilogx()
```

```
axes[1, 0].set_title('Plot 1',
                    fontsize = 40,
                    color = 'red')
```

```
axes[1, 0].legend()
```

Plot 1

```
axes[1, 0].plot([1,2,3], [4,5,6],
               color = 'b',
               linestyle = '--',
               linewidth = 3,
               label = 'first_line')
axes[1, 0].plot([3,2,1,6], [4,8,2,3],
               color='r',
               marker='x',
               markersize = 10,
               linewidth = 0)
```

Label on the legend

Options: .osP+xx

```
axes[1, 0].errorbar([3,2,1,6], [4,8,2,3],
                  yerr = [1, 2, 1, 1],
                  fmt = 'none',
                  capsize = 5)
```

Only plot error bars - do not join them with lines

Size of the bits at the end of the error bars

```
axes[1, 0].set_xlabel('X-Axis  $x^2$ ',
                     fontsize = 20)
axes[1, 0].set_xlim(0, 10)
```

X-Axis x^2

```
axes[1, 1].bar(range(4),
               height = [1,2,3,4],
               width = 0.2,
               align = 'center',
               alpha = 0.5)
axes[1,1].bar([i+0.3 for i in range(4)],
               height = [0.5, 1, 1, 3],
               width = 0.2,
               align = 'center',
               color = 'g')
axes[1,1].bar([i+0.3 for i in range(len(cats))],
               bottom = [0.5, 1, 1, 3],
               height = [0.5, 1, 2, 2],
               width = 0.2,
               align = 'center',
               color = 'r')
```

Put the center of the bars at these points

Bar transparency

Offset the bars

```
axes[1,2].pie(x = [0.05, 0.95],
              labels = ['Pie charts good',
                       'Pie charts bad'],
              labeldistance = 0.5)
```

Pie charts bad

Pie charts good

Set position of the ticks

```
axes[1,1].set_xticks([i + 0.15 for i in range(len(cats))])
axes[1,1].set_xticklabels(['a', 'b', 'c', 'd'],
                           fontsize = 20)
```

Set labels on the ticks

Close out

```
# Make room for everything
fig.tight_layout()

# Save image to PDF
fig.savefig('f.pdf',
           bbox_inches='tight')
```