Matplotlib Example Code:

**Using Numpy Histogram 2d**

|  |
| --- |
| **from** **matplotlib** **import** pyplot **as** plt  hist, xedges, yedges = np.histogram2d(x,y)  X,Y = np.meshgrid(xedges,yedges)  plt.imshow(hist)  plt.grid(**True**)  plt.colorbar()  plt.show() |

Using pcolor from matplotlib

|  |
| --- |
| plt.pcolor(hist)  plt.colorbar()  plt.grid()  plt.show() |

### Using matshow from matplotlib

|  |
| --- |
| **import** **numpy** **as** **np**  **import** **matplotlib.pyplot** **as** **plt**  columns = ['A', 'B', 'C', 'D']  rows = ['1', '2', '3', '4']  data = np.random.random((4,4))  fig = plt.figure()  ax = fig.add\_subplot(111)  cax = ax.matshow(data, interpolation='nearest')  fig.colorbar(cax)  ax.set\_xticklabels([''] + columns)  ax.set\_yticklabels([''] + rows)  plt.show() |

### Diffirent Color Maps

|  |
| --- |
| **from** **math** **import** ceil  **import** **numpy** **as** **np**  *# Sample from a bivariate Gaussian distribution*  mean = [0,0]  cov = [[0,1],[1,0]]  x, y = np.random.multivariate\_normal(mean, cov, 10000).T  size = len(plt.cm.datad.keys())  all\_maps = list(plt.cm.datad.keys())  fig, ax = plt.subplots(ceil(size/4), 4, figsize=(12,100))  counter = 0  **for** row **in** ax:  **for** col **in** row:  **try**:  col.imshow(hist, cmap=all\_maps[counter])  col.set\_title(all\_maps[counter])  **except** **IndexError**:  **break**  counter += 1  plt.tight\_layout()  plt.show() |

### Creating Heat Maps

|  |
| --- |
| **import** **matplotlib.pyplot** **as** **plt**  **from** **matplotlib.colors** **import** LogNorm  **import** **numpy** **as** **np**  np.random.seed(1)  a = np.random.random((25, 25))  plt.subplot(1, 1, 1)  plt.pcolor(a, norm=LogNorm(vmin=a.min() / 1.2, vmax=a.max() \* 1.2), cmap='PuBu\_r')  plt.colorbar()  plt.show() |

### BAR PLOTS

### Bar plot with Error bars

|  |
| --- |
| **import** **matplotlib.pyplot** **as** **plt**  *# input data*  Mean\_values = [1, 2, 3]  variance = [0.2, 0.4, 0.5]  bar\_labels = ['bar 1', 'bar 2', 'bar 3']  *# plot bars*  x\_pos = list(range(len(bar\_labels)))  plt.bar(x\_pos, mean\_values, yerr=variance, align='center', alpha=0.5)  plt.grid()  *# set height of the y-axis*  max\_y = max(zip(mean\_values, variance)) *# returns a tuple, here: (3, 5)*  plt.ylim([0, (max\_y[0] + max\_y[1]) \* 1.1])  *# set axes labels and title*  plt.ylabel('variable y')  plt.xticks(x\_pos, bar\_labels)  plt.title('Bar plot with error bars')  plt.show()  *#plt.savefig('./my\_plot.png')* |

### Grouped Bar Plot

|  |
| --- |
| **import** **matplotlib.pyplot** **as** **plt**  *# Input data*  green\_data = [1, 2, 3]  blue\_data = [3, 2, 1]  red\_data = [2, 3, 3]  labels = ['group 1', 'group 2', 'group 3']  *# Setting the positions and width for the bars*  pos = list(range(len(green\_data)))  width = 0.2    *# Plotting the bars*  fig, ax = plt.subplots(figsize=(8,6))  plt.bar(pos, green\_data, width,  alpha=0.5,  color='g',  label=labels[0])  plt.bar([p + width **for** p **in** pos], blue\_data, width,  alpha=0.5,  color='b',  label=labels[1])    plt.bar([p + width\*2 **for** p **in** pos], red\_data, width,  alpha=0.5,  color='r',  label=labels[2])  *# Setting axis labels and ticks*  ax.set\_ylabel('y-value')  ax.set\_title('Grouped bar plot')  ax.set\_xticks([p + 1.5 \* width **for** p **in** pos])  ax.set\_xticklabels(labels)  *# Setting the x-axis and y-axis limits*  plt.xlim(min(pos)-width, max(pos)+width\*4)  plt.ylim([0, max(green\_data + blue\_data + red\_data) \* 1.5])  *# Adding the legend and showing the plot*  plt.legend(['green', 'blue', 'red'], loc='upper left')  plt.grid()  plt.show() |

# Bar plot with plot labels/text 1

|  |
| --- |
| **from** **matplotlib** **import** pyplot **as** plt  **import** **numpy** **as** **np**  data = range(200, 225, 5)  bar\_labels = ['a', 'b', 'c', 'd', 'e']  fig = plt.figure(figsize=(10,8))  *# plot bars*  y\_pos = np.arange(len(data))  plt.yticks(y\_pos, bar\_labels, fontsize=16)  bars = plt.barh(y\_pos, data,  align='center', alpha=0.4, color='g')  *# annotation and labels*  **for** b,d **in** zip(bars, data):  plt.text(b.get\_width() + b.get\_width()\*0.08, b.get\_y() + b.get\_height()/2,  '**{0:.2%}**'.format(d/min(data)),  ha='center', va='bottom', fontsize=12)  plt.xlabel('X axis label', fontsize=14)  plt.ylabel('Y axis label', fontsize=14)  t = plt.title('Bar plot with plot labels/text', fontsize=18)  plt.ylim([-1,len(data)+0.5])  plt.vlines(min(data), -1, len(data)+0.5, linestyles='dashed')  plt.grid()  plt.show() |

### Bar Plot with Plot labels text2

|  |
| --- |
| **import** **matplotlib.pyplot** **as** **plt**  *# input data*  mean\_values = [1, 2, 3]  bar\_labels = ['bar 1', 'bar 2', 'bar 3']  *# plot bars*  x\_pos = list(range(len(bar\_labels)))  rects = plt.bar(x\_pos, mean\_values, align='center', alpha=0.5)  *# label bars*  **def** autolabel(rects):  **for** ii,rect **in** enumerate(rects):  height = rect.get\_height()  plt.text(rect.get\_x()+rect.get\_width()/2., 1.02\*height, '**%s**'% (mean\_values[ii]),  ha='center', va='bottom')  autolabel(rects)    *# set height of the y-axis*  max\_y = max(zip(mean\_values, variance)) *# returns a tuple, here: (3, 5)*  plt.ylim([0, (max\_y[0] + max\_y[1]) \* 1.1])  *# set axes labels and title*  plt.ylabel('variable y')  plt.xticks(x\_pos, bar\_labels)  plt.title('Bar plot with labels')  plt.show()  *#plt.savefig('./my\_plot.png')* |

# Standard Deviation, Standard Error, and Confidence Intervals

|  |
| --- |
| **import** **numpy** **as** **np**  **from** **matplotlib** **import** pyplot **as** plt  **from** **scipy.stats** **import** t  *# Generating 15 random data points in the range 5-15 (inclusive)*  X = np.random.randint(5, 15, 15)  *# sample size*  n = X.size  *# mean*  X\_mean = np.mean(X)  *# standard deviation*  X\_std = np.std(X)  *# standard error*  X\_se = X\_std / np.sqrt(n)  *# alternatively:*  *# from scipy import stats*  *# stats.sem(X)*  *# 95% Confidence Interval*  dof = n - 1 *# degrees of freedom*  alpha = 1.0 - 0.95  conf\_interval = t.ppf(1-alpha/2., dof) \* X\_std\*np.sqrt(1.+1./n)  fig = plt.gca()  plt.errorbar(1, X\_mean, yerr=X\_std, fmt='-o')  plt.errorbar(2, X\_mean, yerr=X\_se, fmt='-o')  plt.errorbar(3, X\_mean, yerr=conf\_interval, fmt='-o')  plt.xlim([0,4])  plt.ylim(X\_mean-conf\_interval-2, X\_mean+conf\_interval+2)  *# axis formatting*  fig.axes.get\_xaxis().set\_visible(**False**)  fig.spines["top"].set\_visible(**False**)  fig.spines["right"].set\_visible(**False**)  plt.tick\_params(axis="both", which="both", bottom="off", top="off",  labelbottom="on", left="on", right="off", labelleft="on")  plt.legend(['Standard Deviation', 'Standard Error', 'Confidence Interval'],  loc='upper left',  numpoints=1,  fancybox=**True**)  plt.ylabel('random variable')  plt.title('15 random values in the range 5-15')  plt.show() |

# Adding error bars to a barplot

|  |
| --- |
| **import** **matplotlib.pyplot** **as** **plt**  *# input data*  mean\_values = [1, 2, 3]  variance = [0.2, 0.4, 0.5]  bar\_labels = ['bar 1', 'bar 2', 'bar 3']  fig = plt.gca()  *# plot bars*  x\_pos = list(range(len(bar\_labels)))  plt.bar(x\_pos, mean\_values, yerr=variance, align='center', alpha=0.5)  *# set height of the y-axis*  max\_y = max(zip(mean\_values, variance)) *# returns a tuple, here: (3, 5)*  plt.ylim([0, (max\_y[0] + max\_y[1]) \* 1.1])  *# set axes labels and title*  plt.ylabel('variable y')  plt.xticks(x\_pos, bar\_labels)  plt.title('Bar plot with error bars')  *# axis formatting*  fig.axes.get\_xaxis().set\_visible(**False**)  fig.spines["top"].set\_visible(**False**)  fig.spines["right"].set\_visible(**False**)  plt.tick\_params(axis="both", which="both", bottom="off", top="off",  labelbottom="on", left="on", right="off", labelleft="on")  plt.show() |