My First Plot

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.gridspec as gridspec
#Write your code here
def test_my_first_plot():
  fig = plt.figure(figsize=(8,6))
  ax = fig.add_subplot(111)
  t = [5, 10, 15, 20, 25]
  d = [25, 50, 75, 100, 125]
  ax.set(title='Time vs Distance Covered',
        xlabel='time (seconds)', ylabel='distance (meters)',
        xlim=(0, 30), ylim=(0,130))
  plt.plot(t, d, label='d = 5t')
  plt.legend()
  plt.savefig('scatter.png')
test_my_first_plot()
```

Create Line and Scatter Plots

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.gridspec as gridspec
import matplotlib.ticker as ticker
#Write your code here
def test_sine_wave_plot():
  fig = plt.figure(figsize=(12,3))
  ax = fig.add_subplot(111)
  t = np.linspace(0.0,2.0,200)
  v = np.sin(2.5*np.pi*t)
  plt.plot(t, v, label='sin(t)',color = 'red')
  ax.set(title='Sine Wave', xlabel='Time (seconds)', ylabel='Voltage (mV)',  xlim=(0, 2), ylim=(-1,1))
  xmajor = [0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8,2.0]
  ymajor = [-1,0,1]
  ax.xaxis.set_major_formatter(ticker.FixedFormatter(xmajor))
  ax.yaxis.set_major_formatter(ticker.FixedFormatter(ymajor))
  plt.grid(linestyle='--')
  plt.legend()
  plt.savefig('sinewave.png')
test_sine_wave_plot()

def test_multi_curve_plot():
  fig = plt.figure(figsize=(12,3))
  ax = fig.add_subplot(111)
  x = np.linspace(0.0,5.0,20)
  y1 = x
  y2 = x**2
  y3 = x**3
  ax.set(title='Linear, Quadratic, & Cubic Equations', xlabel='X', ylabel='f(x)')
  ax.plot(x, y1, label='y=x',marker = 'o',color = 'red')
  ax.plot(x, y2, label='y = x**2',marker = "s",color = 'green')
  ax.plot(x, y3, label='y = x**3',marker = "^",color = 'blue')
  plt.legend()
  plt.savefig('multicurve.png')
test_multi_curve_plot()

def test_scatter_plot():
  fig = plt.figure(figsize=(12,3))
  ax = fig.add_subplot(111)
  s = [50, 60, 55, 50, 70, 65, 75, 65, 80, 90, 93, 95]
  months = [1,2,3,4,5,6,7,8,9,10,11,12]
  ax.set(title="Cars Sold by Company 'X' in 2017", xlabel='Months', ylabel='No. of Cars Sold',xlim=(0, 13),
ylim=(20,100))
  ax.scatter(months, s, marker = 'o', color = 'red')
  plt.xticks([1, 3, 5, 7, 9,11])
  ax.set_xticklabels(['Jan', 'Mar', 'May', 'Jul', 'Sep', 'Nov'])
  plt.savefig('scatter.png')
test_scatter_plot()
```

3 Bar Plots

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.gridspec as gridspec
#Write your code here
def test_barplot_of_iris_sepal_length():
  fig = plt.figure(figsize = (8,6))
  ax = fig.add_subplot(111)
  species = ['setosa', 'versicolor', 'viriginica']
  index = [0.2, 1.2, 2.2]
  sepal_len = [5.01, 5.94, 6.59]
  ax.set(title = "Mean Sepal Length of Iris Species",
  xlabel = "Specices",ylabel = "Sepal Length (cm)",
  xlim = (0,3),ylim = (0,7))
  ax.bar(index,sepal_len,width = 0.5,color = "red",edgecolor = "black")
  plt.xticks([0.45,1.45,2.45])
  ax.set_xticklabels(['setosa', 'versicolor', 'viriginica'])
  ax.legend()
  plt.savefig('bar_iris_sepal.png')
test_barplot_of_iris_sepal_length()

def test_barplot_of_iris_measurements():
  fig = plt.figure(figsize = (8,6))
  ax = fig.add_subplot(111)
  sepal_len = [5.01, 5.94, 6.59]
  sepal_wd = [3.42, 2.77, 2.97]
  petal_len = [1.46, 4.26, 5.55]
  petal_wd = [0.24, 1.33, 2.03]
  species = ['setosa', 'versicolor', 'viriginica']
  species_index1 = [0.7, 1.7, 2.7]
  species_index2 = [0.9, 1.9, 2.9]
  species_index3 = [1.1, 2.1, 3.1]
  species_index4 = [1.3, 2.3, 3.3]
  ax.set(title = "Mean Measurements of Iris Species",xlabel = "Specices",ylabel = "Iris Measurements
(cm)",
      xlim = (0.5,3.7),ylim = (0,10))
  ax.bar(species_index1, sepal_len, color='c', width=0.2, edgecolor='black', label='Sepal Length')
  ax.bar(species_index2, sepal_wd, color='m', width=0.2, edgecolor='black', label='Sepal Width')
  ax.bar(species_index3, petal_len, color='y', width=0.2, edgecolor='black', label='Petal Length')
  ax.bar(species_index4, petal_wd, color='orange', width=0.2, edgecolor='black', label='Petal Width')
  ax.set_xticks([1.1,2.1,3.1])
  ax.set_xticklabels(['setosa', 'versicolor','viriginica'])
  ax.legend()
  plt.savefig('bar_iris_measure.png')

test_barplot_of_iris_measurements()

def test_hbarplot_of_iris_petal_length():
  fig = plt.figure(figsize = (12,5))
  ax = fig.add_subplot(111)
  species = ['setosa', 'versicolor', 'viriginica']
  index = [0.2, 1.2, 2.2]
  petal_len = [1.46, 4.26, 5.55]
  ax.set(title = "Mean Petal Length of Iris Species",ylabel = "Specices",xlabel = "Petal Length (cm)")
  ax.barh(index, petal_len, color='c', height = 0.5, edgecolor='black', label='Sepal Length')
  ax.set_yticks([0.45, 1.45,2.45])
  ax.set_yticklabels(['setosa', 'versicolor','viriginica'])
  ax.legend()
  plt.savefig('bar_iris_petal.png')

test_hbarplot_of_iris_petal_length()
```

4 Histograms and Box Plots

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.gridspec as gridspec
#Write your code here
def test_hist_of_a_sample_normal_distribution():
  fig = plt.figure(figsize = (8,6))
  ax = fig.add_subplot(111)
  np.random.seed(100)
  x1 = 25+3*np.random.randn(1000)
  ax.set(title = "Histogram of a Single Dataset",xlabel = "x1",ylabel = "Bin Count")
  ax.hist(x1,bins = 30)
  plt.savefig('histogram_normal.png')

test_hist_of_a_sample_normal_distribution()

def test_boxplot_of_four_normal_distribution():
  fig = plt.figure(figsize = (8,6))
  ax = fig.add_subplot(111)
  np.random.seed(100)
  x1 = 25+3.0*np.random.randn(1000)
  x2 = 35+5.0*np.random.randn(1000)
  x3 = 55+10.0*np.random.randn(1000)
  x4 = 45+3.0*np.random.randn(1000)
  ax.set(title = "Box plot of Multiple Datasets",xlabel = "Dataset",ylabel = "Value")
  ax.boxplot([x1, x2, x3, x4], labels=['X1', 'X2', 'X3', 'X4'], notch=True,patch_artist = "True",showfliers  =
"+")
  plt.savefig('box_distribution.png')

test_boxplot_of_four_normal_distribution()
```

5 Applying styles

```
def test_generate_plot_with_style1():
  fig = plt.figure(figsize = (8,6))
  ax = fig.add_subplot(111)
  sepal_len = [5.01, 5.94, 6.59]
  sepal_wd = [3.42, 2.77, 2.97]
  petal_len = [1.46, 4.26, 5.55]
  petal_wd = [0.24, 1.33, 2.03]
  species = ['setosa', 'versicolor', 'viriginica']
  species_index1 = [0.7, 1.7, 2.7]
  species_index2 = [0.9, 1.9, 2.9]
  species_index3 = [1.1, 2.1, 3.1]
  species_index4 = [1.3, 2.3, 3.3]
  ax.set(title = "Mean Measurements of Iris Species",xlabel = "Specices",ylabel = "Iris Measurements
(cm)",
       xlim = (0.5,3.7),ylim = (0,10))
  ax.bar(species_index1, sepal_len, color='c', width=0.2, edgecolor='black', label='Sepal Length')
  ax.bar(species_index2, sepal_wd, color='m', width=0.2, edgecolor='black', label='Sepal Width')
  ax.bar(species_index3, petal_len, color='y', width=0.2, edgecolor='black', label='Petal Length')
  ax.bar(species_index4, petal_wd, color='orange', width=0.2, edgecolor='black', label='Petal Width')
  ax.set_xticks([1.1,2.1,3.1])
  ax.set_xticklabels(['setosa', 'versicolor','viriginica'])
  ax.legend()
  plt.savefig('plotstyle1.png')
test_generate_plot_with_style1()
```

6 Multiple Plots

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.gridspec as gridspec
#Write your code here
def test_generate_figure1():
  fig = plt.figure(figsize=(8,6))
  axes1 = plt.subplot(2, 1, 1, title='Sin(2*pi*x)')
  axes2 = plt.subplot(2, 1, 2, title='Sin(4*pi*x)',sharex=axes1, sharey=axes1)
  t = np.arange(0.0, 5.0, 0.01)
  s1 = np.sin(2*np.pi*t)
  s2 = np.sin(4*np.pi*t)
  axes1.plot(t, s1)
  axes2.plot(t, s2)
  plt.savefig('testfigure1.png')
test_generate_figure1()

def test_generate_figure2():
  fig = plt.figure(figsize = (8,6))
  axes1 = plt.subplot(2,2,1,title = "Scatter plot with Upper Traingle Markers")
  plt.xticks([0.0, 0.4, 0.8, 1.2])
  plt.yticks([-0.2, 0.2, 0.6, 1.0])
  axes2 = plt.subplot(2,2,2,title = "Scatter plot with Plus Markers")
  plt.xticks([0.0, 0.4, 0.8, 1.2])
  plt.yticks([-0.2, 0.2, 0.6, 1.0])
  axes3 = plt.subplot(2,2,3,title = "Scatter plot with Circle Markers")
  plt.xticks([0.0, 0.4, 0.8, 1.2])
  plt.yticks([-0.2, 0.2, 0.6, 1.0])
  axes4 = plt.subplot(2,2,4,title = "Scatter plot with Diamond Markers")
  plt.xticks([0.0, 0.4, 0.8, 1.2])
  plt.yticks([-0.2, 0.2, 0.6, 1.0])
  np.random.seed(1000)
  x = np.random.rand(10)
  y = np.random.rand(10)
  z = np.sqrt(x**2+y**2)
  axes1.scatter(x,y, s = 80,c =z,marker = "^")
  axes2.scatter(x,y, s = 80,c =z,marker = "+")
  axes3.scatter(x,y, s = 80,c =z,marker = "o")
  axes4.scatter(x,y, s = 80,c =z,marker = "d")
  plt.tight_layout()
  plt.savefig('testfigure2.png')
test_generate_figure2()

def test_generate_figure3():
  fig = plt.figure(figsize=(8,16))
  g = gridspec.GridSpec(2,2)
  x = np.arange(0,101)
  y1 = x
  y2 = x**2
  y3 = x**3
  axes1 = plt.subplot(g[0,:1],title='y = x')
  axes2 = plt.subplot(g[1,:1], title='y = x**2')
  axes3 = plt.subplot(g[:, 1], title='y = x**3')
  axes1.plot(x, y1)
  axes2.plot(x, y2)
  axes3.plot(x, y3)
  plt.tight_layout()
  plt.savefig('testfigure3.png')
test_generate_figure3()
```