NOTE: ALL THE COMMANDS FOR PLOTTING A FIGURE SHOULD ALL GO IN THE SAME CELL. SEPARATING THEM OUT INTO MULTIPLE CELLS MAY CAUSE NOTHING TO SHOW UP.

Exercises

Follow the instructions to recreate the plots using this data:

Data

```
import numpy as np
x = np.arange(0,100)
y = x*2
z = x**2
```

Import matplotlib.pyplot as plt and set %matplotlib inline if you are using the jupyter notebook. What command do you use if you aren't using the jupyter notebook?

Double-click (or enter) to edit

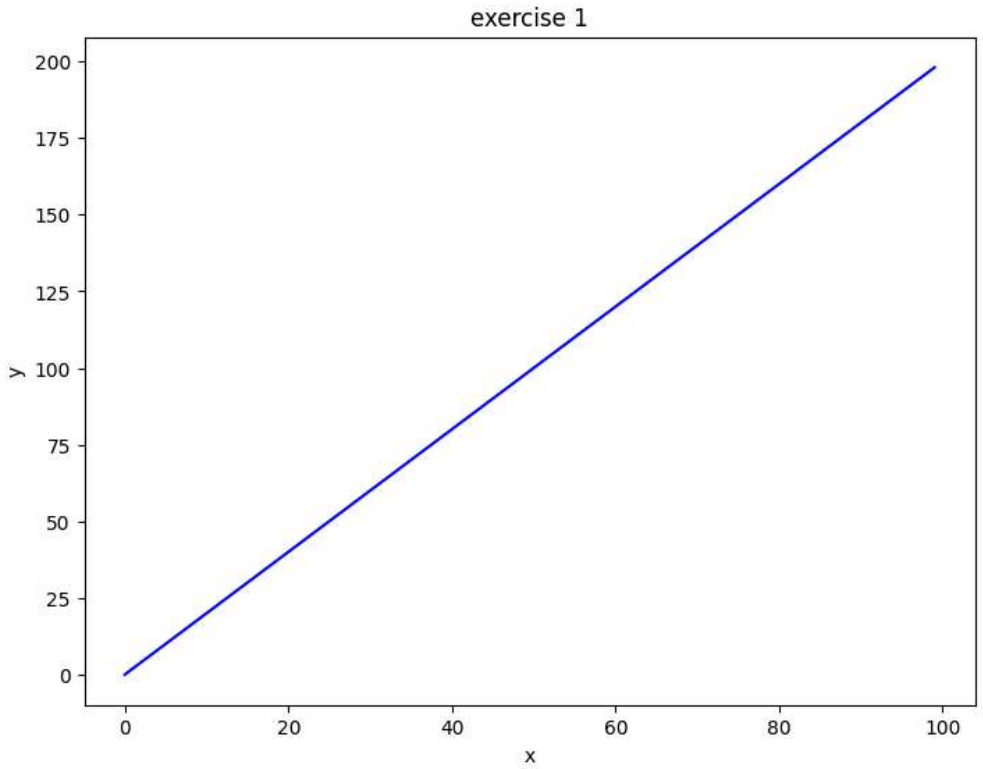
```
import matplotlib.pyplot as plt
```

Exercise 1

Follow along with these steps

- Create a figure object called fig using plt.figure()
- Use add_axes to add an axis to the figure canvas at [0,0,1,1]. Call this new axis ax.
- Plot (x,y) on that axes and set the labels and titles to match the plot below:

```
fig=plt.figure()
ax=fig.add_axes([0,0,1,1])
ax.plot(x,y,'b-')
ax.set_xlabel('x')          # X-axis label
ax.set_ylabel('y')          # Y-axis label
ax.set_title('exercise 1')
plt.show()
```

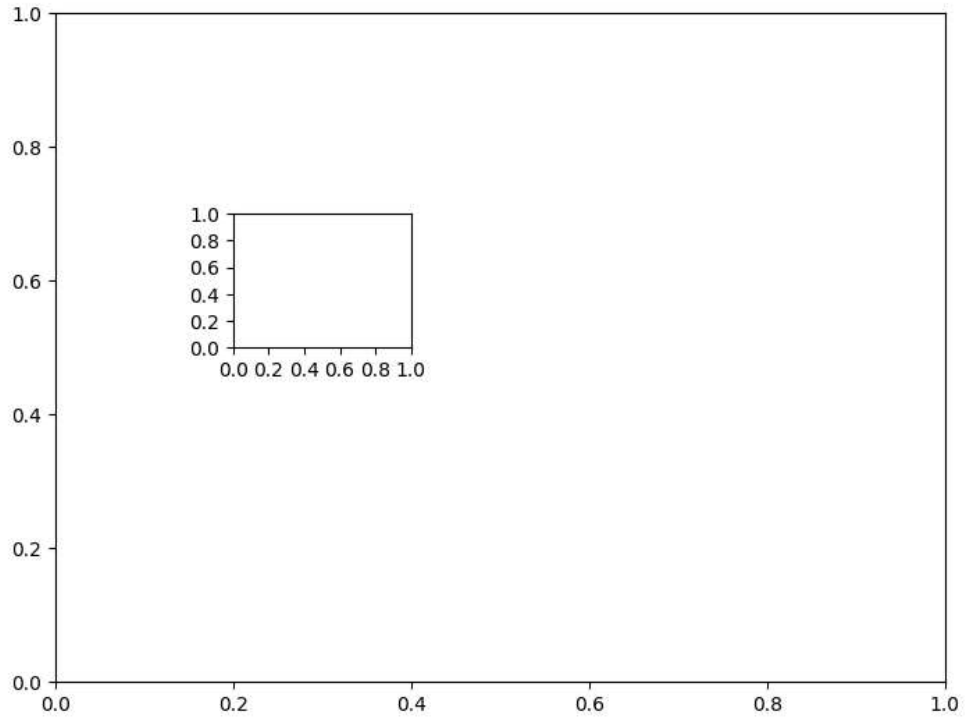


Exercise 2

Create a figure object and put two axes on it, ax1 and ax2. Located at [0,0,1,1] and [0.2,0.5,.2,.2] respectively.

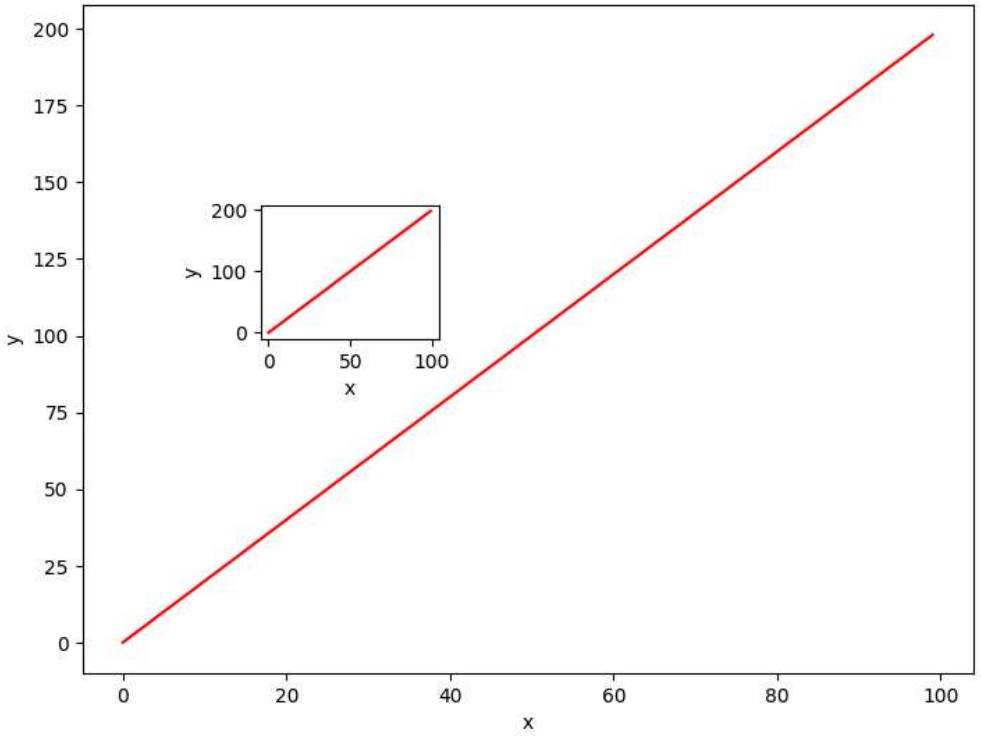
```
fig = plt.figure()
ax1 = fig.add_axes([0,0,1,1])
ax2 = fig.add_axes([0.2,0.5,0.2,0.2])
ax2.set_xticks([0,0.2, 0.4, 0.6,0.8,1.0])
ax2.set_yticks([0,0.2, 0.4, 0.6,0.8,1.0])
```

```
[<matplotlib.axis.YTick at 0x7a5e391591f0>,
 <matplotlib.axis.YTick at 0x7a5e39158cb0>,
 <matplotlib.axis.YTick at 0x7a5e39180b30>,
 <matplotlib.axis.YTick at 0x7a5e3915e9f0>,
 <matplotlib.axis.YTick at 0x7a5e39181640>,
 <matplotlib.axis.YTick at 0x7a5e391820f0>]
```



Now plot (x,y) on both axes. And call your figure object to show it.

```
fig = plt.figure()
ax1 = fig.add_axes([0,0,1,1])
ax2 = fig.add_axes([0.2,0.5,0.2,0.2])
ax1.plot(x,y,'r-')
ax1.set_xlabel('x')
ax1.set_ylabel('y')
ax2.plot(x,y,'r-')
ax2.set_xlabel('x')           # X-axis label
ax2.set_ylabel('y')
plt.show()          # Y-axis label          # Y-axis label
```



Exercise 3

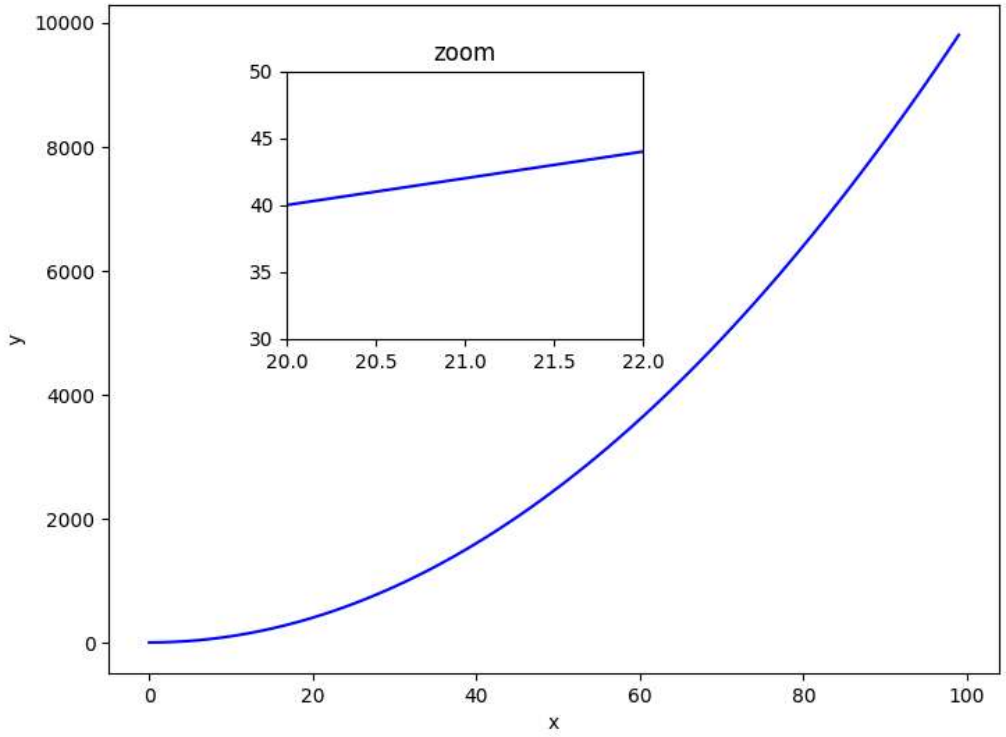
Create the plot below by adding two axes to a figure object at [0,0,1,1] and [0.2,0.5,.4,.4]

```
fig2 = plt.figure()
ax1 = fig.add_axes([0,0,1,1])
ax2 = fig.add_axes( [0.2,0.5,.4,.4])
ax2.set_xticks([0,0.2, 0.4, 0.6,0.8,1.0])
ax2.set_yticks([0,0.2, 0.4, 0.6,0.8,1.0])
```

```
[<matplotlib.axis.YTick at 0x7a5e3b4a6ff0>,
 <matplotlib.axis.YTick at 0x7a5e3b4a64e0>,
 <matplotlib.axis.YTick at 0x7a5e3b4f8380>,
 <matplotlib.axis.YTick at 0x7a5e3b4f8dd0>,
 <matplotlib.axis.YTick at 0x7a5e3b48e750>,
 <matplotlib.axis.YTick at 0x7a5e3b4f9760>]
<Figure size 640x480 with 0 Axes>
```

Now use x,y, and z arrays to recreate the plot below. Notice the xlimits and y limits on the inserted plot:

```
fig3=plt.figure()
ax1=fig3.add_axes([0,0,1,1])
ax1.plot(x,z,'b-')
ax1.set_xlabel('x')           # X-axis label
ax1.set_ylabel('z')
ax2=fig3.add_axes([0.2,0.5,.4,.4])
ax2.plot(x,y,'b-')
ax1.set_xlabel('x')           # X-axis label
ax1.set_ylabel('y')
ax2.set_xlim(20,22)
ax2.set_ylim(30,50)
ax2.set_title('zoom')
plt.show()
```



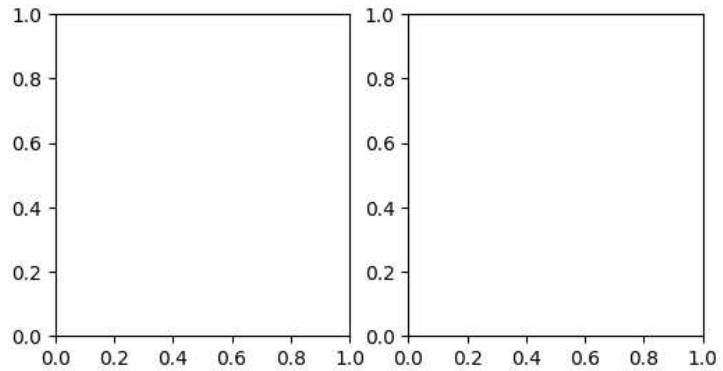
Exercise 4

Use plt.subplots(nrows=1, ncols=2) to create the plot below.

```
fig=plt.figure()
fig,axes=plt.subplots(nrows=1,ncols=2,figsize=(6,3))
axes[0].set_xlim(0, 1)
axes[0].set_ylim(0, 1)
#axes[0].set_xticks[0.0,0.2,0.4,0.6,0.8,1.0]
```

```
axes[1].set_xlim(0, 1)
axes[1].set_ylim(0, 1)
```

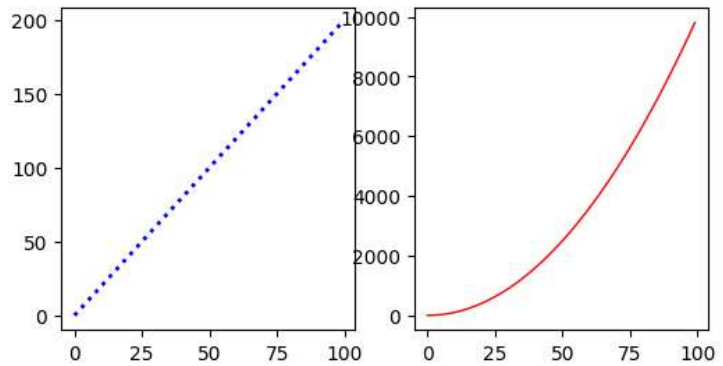
```
(0.0, 1.0)
<Figure size 640x480 with 0 Axes>
```



Now plot (x,y) and (x,z) on the axes. Play around with the linewidth and style

```
fig=plt.figure()
fig,axes=plt.subplots(nrows=1,ncols=2,figsize=(6,3))
axes[0].plot(x,y,'b-',linestyle=':',linewidth=2)
axes[1].plot(x,z,'r-',linewidth=1)
plt.show()
```

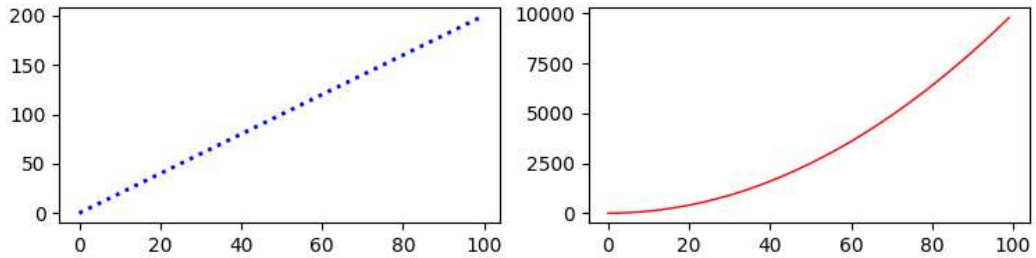
```
/tmp/ipython-input-3850843342.py:3: UserWarning: linestyle is redundantly defined by the 'linestyle' keyword argument and the fmt string "b-" (-> linestyle='-'). The k
  axes[0].plot(x,y,'b-',linestyle=':',linewidth=2)
<Figure size 640x480 with 0 Axes>
```



See if you can resize the plot by adding the figsize() argument in plt.subplots() are copying and pasting your previous code.

```
fig=plt.figure()
fig,axes=plt.subplots(nrows=1,ncols=2,figsize=(9,2))
axes[0].plot(x,y,'b-',linestyle=':',linewidth=2)
axes[1].plot(x,z,'r-',linewidth=1)
plt.show()
```

```
/tmp/ipython-input-2878007762.py:3: UserWarning: linestyle is redundantly defined by the 'linestyle' keyword argument and the fmt string "b-" (-> linestyle='-'). The k
  axes[0].plot(x,y,'b-',linestyle=':',linewidth=2)
<Figure size 640x480 with 0 Axes>
```



#Plots

Plot the following plots using matplotlib, you many generate your own data or get it from the internet:

1. Histogram
2. Scatter Plot
3. Boxplot
4. Pie charts

(Bonus - Read up on the use-case of histograms in EDA and ML in general, they are VERY useful! Also, look up **Violin plots** if you have extra time)

```
import numpy as np
import pandas as pd
fig=plt.figure()
fig,axes=plt.subplots(2,2,figsize=(10,10))
np.random.seed(42)
#for histogram
hist_data = np.random.normal(loc=50, scale=15, size=1000)
axes[0, 0].hist(hist_data, bins=30, alpha=0.7, color='skyblue', edgecolor='black')
axes[0, 0].set_title('Histogram')
#for scatter plot
x_scatter = np.random.randn(100)
y_scatter = 2*x_scatter + np.random.randn(100) * .5
axes[0, 1].scatter(x_scatter, y_scatter, alpha=0.6, color='coral', s=50)
axes[0, 1].set_title('Scatter Plot')
axes[0, 1].set_xlabel('X')
axes[0, 1].set_ylabel('Y')
#for piechart
groups=['never','once in a week','frequently']
```

```
data=np.array([10,20,70])
axes[1, 0].pie(data, labels=groups, autopct='%1.1f%%', startangle=90)
axes[1, 0].set_title('Pie Chart')
#for boxplot
data_normal = np.random.normal(100, 15, 1000)  # Normal distribution
data_skewed = np.random.normal(50,3,100)
axes[1, 1].boxplot([data_normal, data_skewed], labels=['Normal', 'Skewed'])
axes[1, 1].set_title('Boxplot')
```

```
/tmp/ipython-input-1248194166.py:25: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick_labels' since Matplotlib 3.9; support f r
  axes[1, 1].boxplot([data_normal, data_skewed], labels=['Normal', 'Skewed'])
Text(0, 0.5, 'Value')
<Figure size 640x480 with 0 Axes>
```