

INTRO TO PYTHON 1

Be sure to get an account at

<https://research.google.com/colaboratory/>



The screenshot shows the top part of a Jupyter Notebook interface. At the top left is the 'CO' logo. Next to it is the file name 'Untitled5.ipynb' with a star icon to its right. Below the file name is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. To the right of the menu bar is the text 'All changes saved'. Below the menu bar are two buttons: '+ Code' and '+ Text'. Below these buttons is a text input field containing the text 'Start coding or generate with AI.'. On the left side of the interface is a vertical sidebar with several icons: a hamburger menu, a search icon, a keyboard icon, a folder icon, a double arrow icon, a list icon, and a terminal icon. Annotations include: a gold arrow pointing from the file name to the text 'You can change the title of your file'; a gold arrow pointing from the '+ Code' button to the text 'You can add a code cell'; a gold arrow pointing from the '+ Text' button to the text 'You can add a text cell'; and a black arrow pointing from the bottom of the sidebar area to a gold box containing instructions.

CO Untitled5.ipynb ☆ You can change the title of your file

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Start coding or generate with AI.

You can add a code cell

You can add a text cell

All files will be saved in your Google Drive.

TO DO: Download the file intro.ipynb on the website. Use the File dropdown to upload the file



+ Code + Text

To run/compile the entire cell click this arrow or Choose Run all in the Runtime dropdown menu.



```
import pip
import pandas as pd
import numpy as np
import scipy
from numpy import linalg
from sklearn.linear_model import LinearRegression
import statistics as stats
import matplotlib.pyplot as plt
from scipy import stats
import numpy.polynomial.polynomial as poly
```

This is a code cell. It contains a variety of packages that are useful in modeling of data and data visualization.

TO DO: Click on the + Text button to get a Text box.
Type text "Creating a Scatterplot"
Then click the +Code button



+ Code + Text

```
[ ] import pip
import pandas as pd
import numpy as np
import scipy
from numpy import linalg
from sklearn.linear_model import LinearRegression
import statistics as stats
import matplotlib.pyplot as plt
from scipy import stats
import numpy.polynomial.polynomial as poly
```

Creating a Scatterplot

▶ Start coding or generate with AI.

There are lots of ways to input data, we will be using arrays.

Our practice data set will be $\{(1,0),(2,5),(3,4),(6,9)\}$

TO DO: Create your x array. Type $x=np.array([1,2,3,6])$. Note that “x” is a name for this array, you can call it whatever you want.

Hit enter and then create an array for your y values.



+ Code + Text

```
[ ] import pip
import pandas as pd
import numpy as np
import scipy
from numpy import linalg
from sklearn.linear_model import LinearRegression
import statistics as stats
import matplotlib.pyplot as plt
from scipy import stats
import numpy.polynomial.polynomial as poly
```

Creating a Scatterplot

```
▶ x=np.array([1,2,3,6])
y=np.array([0,5,4,9])
```

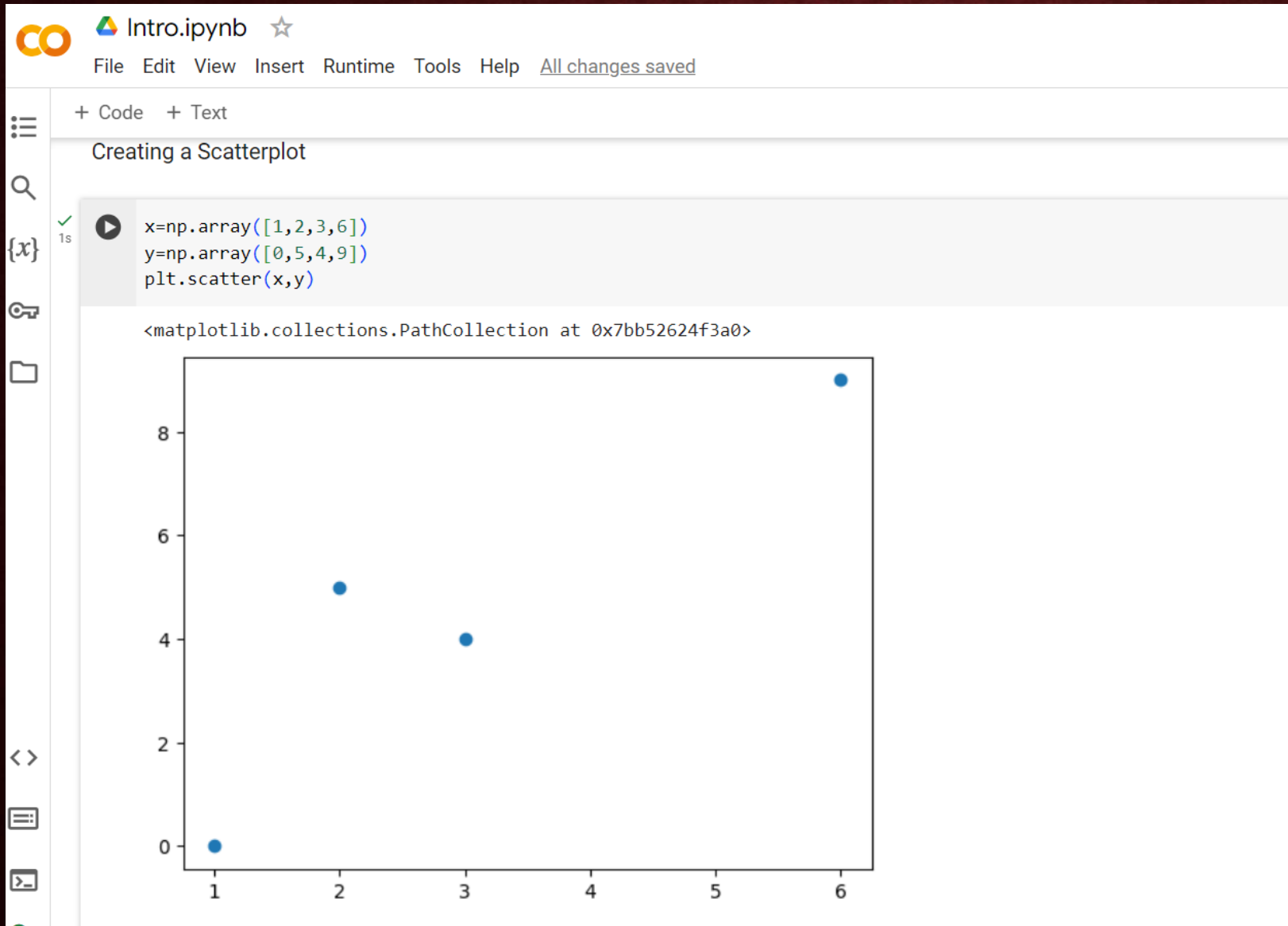
These commands store the data in the variables x and y.

Now let's try a scatterplot.

Type: `plt.scatter(x,y)`

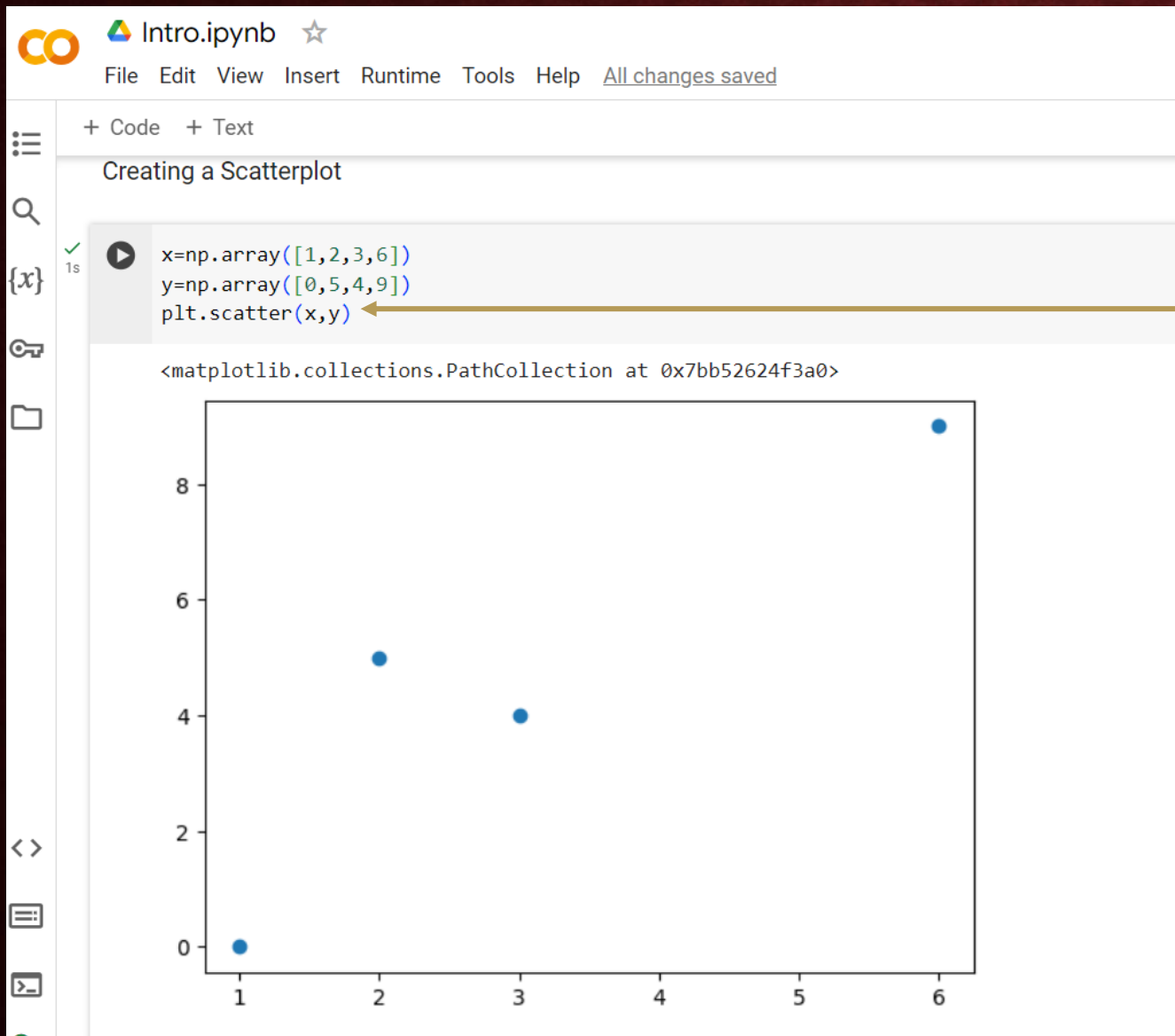
in the same cell and click the arrow.





Didn't get a graph?
If so, too much time
may have passed.

Colab is reliant on
your internet
connection and if it
timed out, or if you
left and came back,
run all of your cells
by using the Runtime
dropdown – Run all



Let's add some axis labels and graph title.

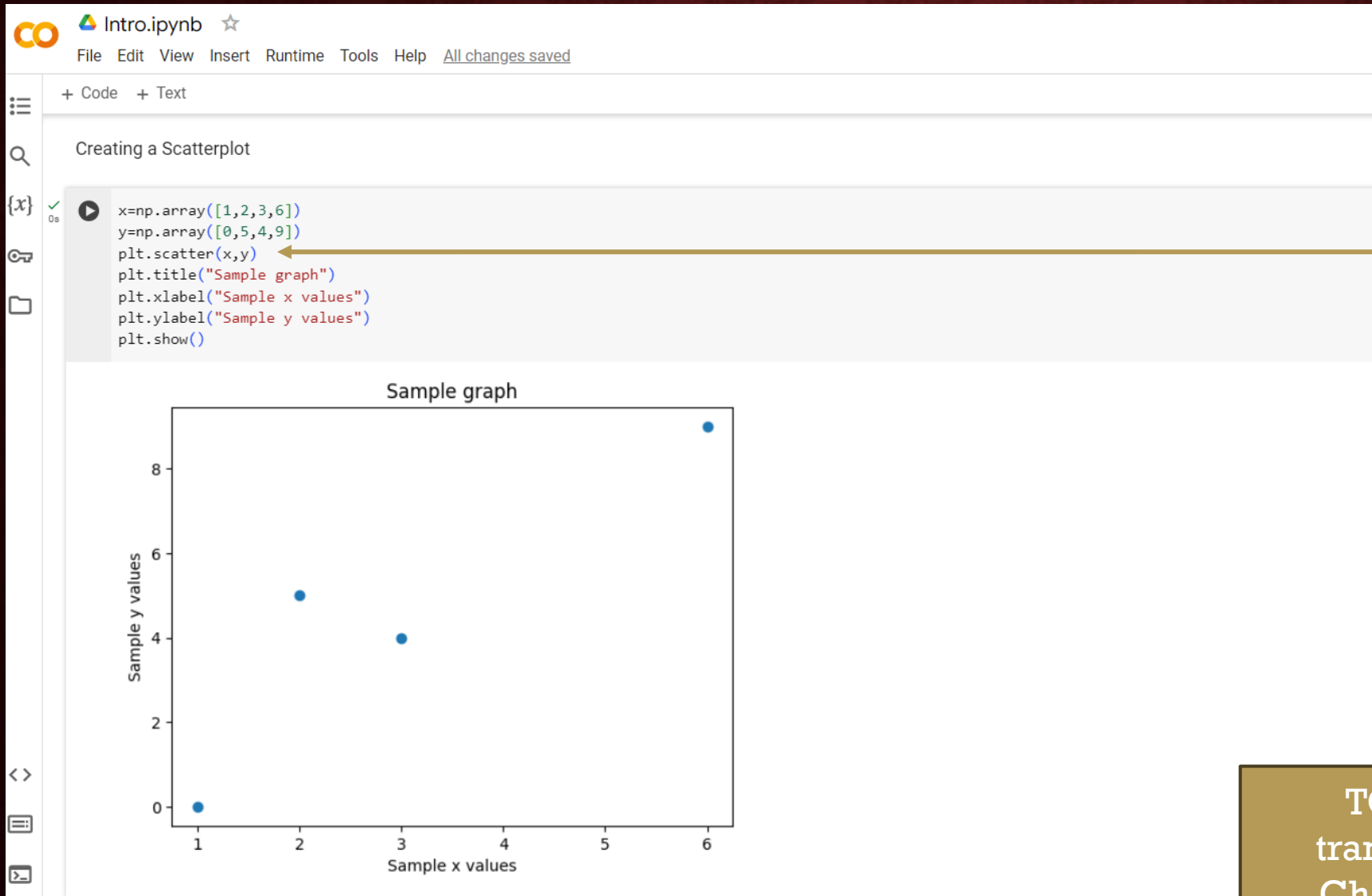
In the same cell, type:

```
plt.title("Name your Graph")  
plt.xlabel("Name your xaxis")  
plt.ylabel("Name your yaxis")  
plt.show()
```

Then Runtime → Runall


Or click





Features of the scatterplot.
Want to change the color or
transparency of the dots

Add it here. Try it:
`plt.scatter(x,y,marker="s",
alpha=.5,color="red")`

Then Runtime → Run all
Or click 

Find more markers here:
[https://matplotlib.org/stable
/api/markers_api.html](https://matplotlib.org/stable/api/markers_api.html)

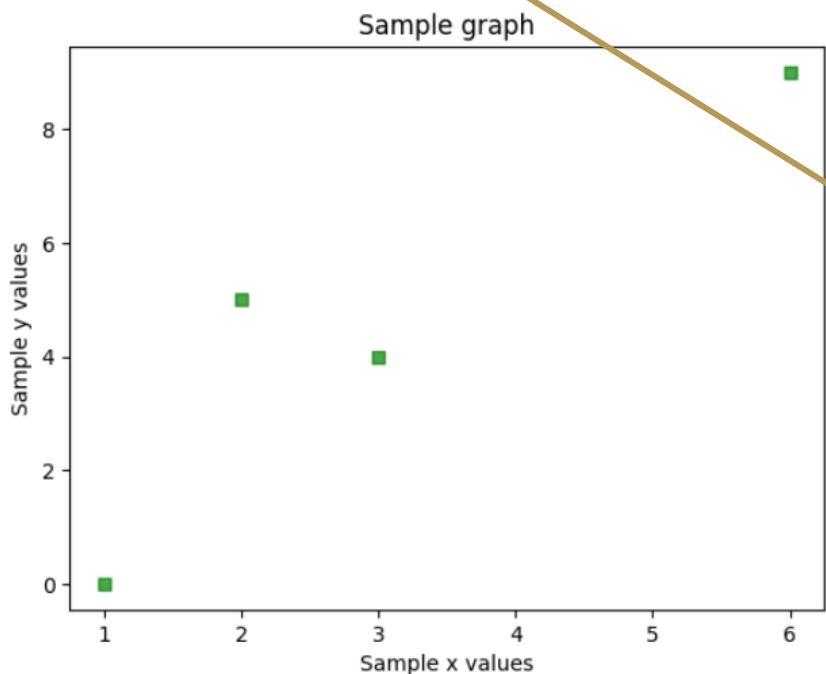
TO DO: Make the dots more
transparent by changing alpha.
Change the color to green and
change the marker to a Diamond.



+ Code + Text

Creating a Scatterplot

```
x=np.array([1,2,3,6])
y=np.array([0,5,4,9])
plt.scatter(x,y,marker="s",alpha=.7,color="green")
plt.title("Sample graph")
plt.xlabel("Sample x values")
plt.ylabel("Sample y values")
plt.show()
```



Want to highlight just one data point with different features?

Python arrays start counting with the number 0.

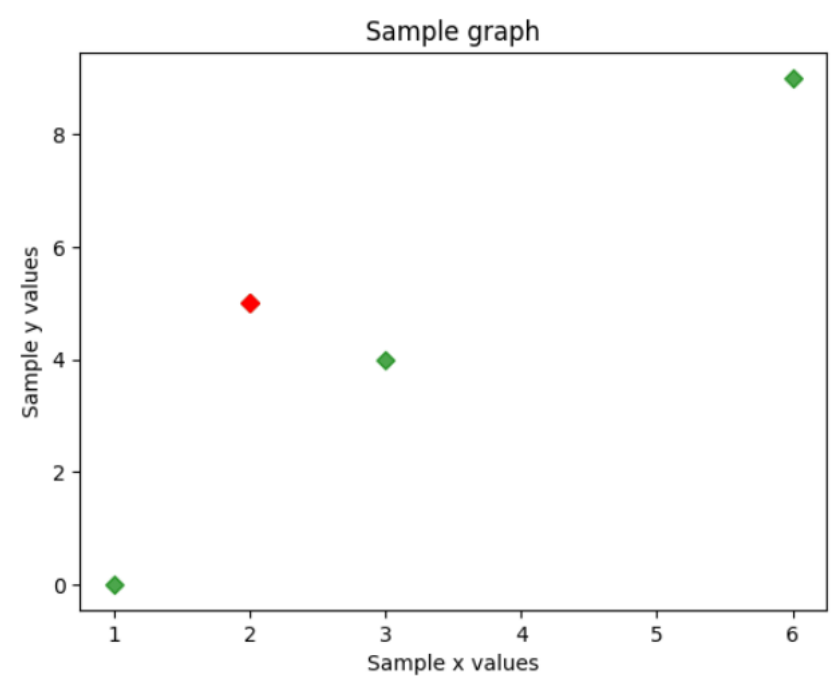
For example $x[0]=1$ and $y[1]=5$.

So `plt.scatter(x[1],y[1])` would plot the point (2,5).

TO DO: Add a `plt.scatter` line where the arrow is pointing to plot the point (2,5) in red as a diamond. Don't forget to run your code!

Creating a Scatterplot

```
x=np.array([1,2,3,6])  
y=np.array([0,5,4,9])  
plt.scatter(x,y,marker="D",alpha=.7,color="green")  
plt.scatter(x[1],y[1],marker="D",color="red")  
plt.title("Sample graph")  
plt.xlabel("Sample x values")  
plt.ylabel("Sample y values")  
plt.show()
```



Want to label your data?

Create annotations

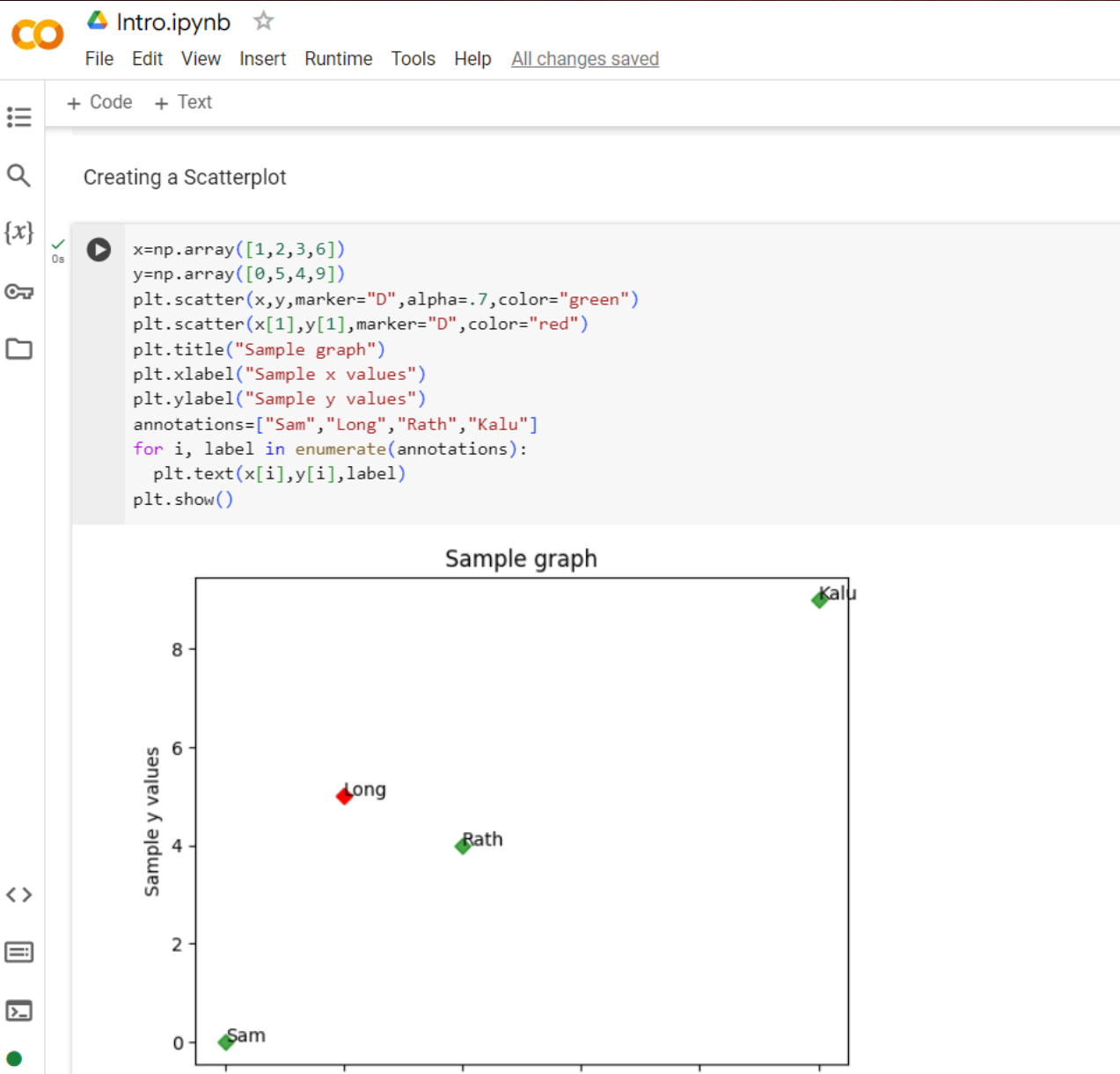
```
annotations=["Sam","Qian","Rath","Kalu"]
```

Then you have to use a for loop to add the labels.
We will talk more about for loops later.

```
for i, label in enumerate(annotations):  
    plt.text(x[i],y[i],label)
```

Indent just one tab

TO DO: Add annotations to your graph



You might want to move your label so that they are not on top of your points.

You can choose how much to move the labels in both the x and y directions, relative to the data.

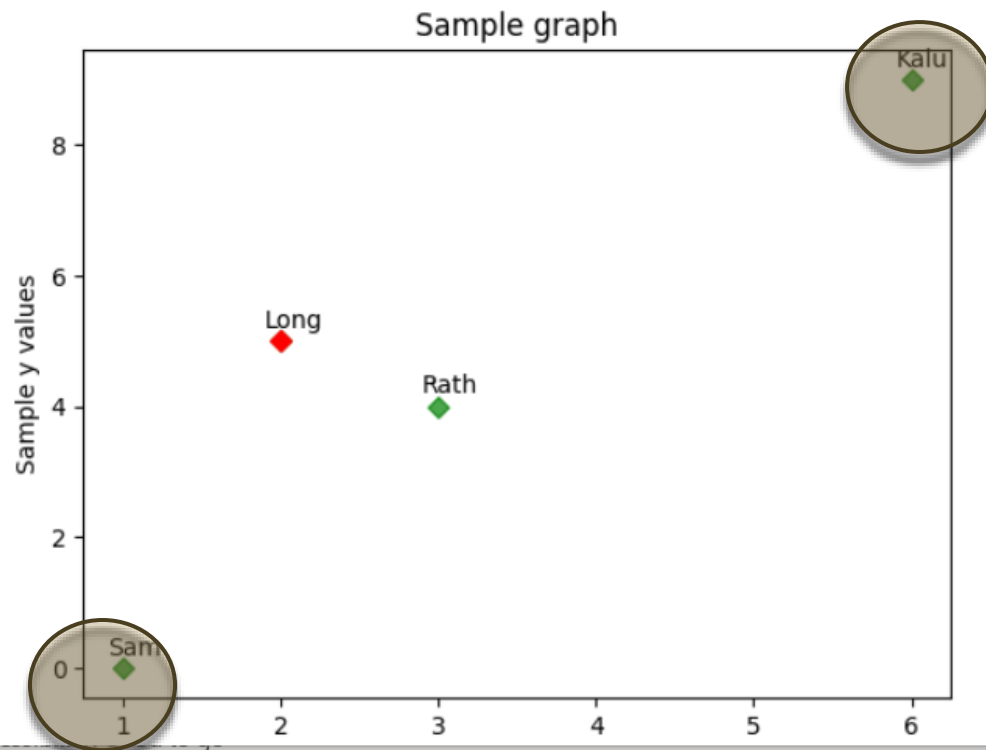
For example, `plt.text(x[i]+.1,y[i]-.02, label)` moves labels right .1 from their dot moves labels down .02 from their dot

TO DO: Move your labels left .1 and up .2

+ Code + Text

Creating a Scatterplot

```
x=np.array([1,2,3,6])
y=np.array([0,5,4,9])
plt.scatter(x,y,marker="D",alpha=.7,color="green")
plt.scatter(x[1],y[1],marker="D",color="red")
plt.title("Sample graph")
plt.xlabel("Sample x values")
plt.ylabel("Sample y values")
annotations=["Sam","Long","Rath","Kalu"]
for i, label in enumerate(annotations):
    plt.text(x[i]-.1,y[i]+.2,label)
plt.show()
```



You might want to change the x range and y range to include (0,0) or make sure you can see your label for Kalu.

```
plt.xlim([minimum x, maximum x])
plt.ylim([minimum y, maximum y])
```

Make sure to put this before `plt.show()`

TO DO: Change your x range to `[0,7]` and y range to `[0,10]`

WATCH THE 2ND PYTHON VIDEO TO LEARN HOW TO IMPORT DATA, AND FIND AND GRAPH A SIMPLE LINEAR REGRESSION MODEL

