In [1]:

```
!pip install matplotlib
```

```
Requirement already satisfied: matplotlib in /Users/mohit/opt/anaconda
3/lib/python3.8/site-packages (3.3.4)
Requirement already satisfied: pillow>=6.2.0 in /Users/mohit/opt/anaco
nda3/lib/python3.8/site-packages (from matplotlib) (8.2.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.
3 in /Users/mohit/opt/anaconda3/lib/python3.8/site-packages (from matp
lotlib) (2.4.7)
Requirement already satisfied: numpy>=1.15 in /Users/mohit/opt/anacond
a3/lib/python3.8/site-packages (from matplotlib) (1.20.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/mohit/opt/a
naconda3/lib/python3.8/site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /Users/mohit/opt/anacon
da3/lib/python3.8/site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in /Users/mohit/op
t/anaconda3/lib/python3.8/site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: six in /Users/mohit/opt/anaconda3/lib/p
ython3.8/site-packages (from cycler>=0.10->matplotlib) (1.15.0)
```

In [19]:

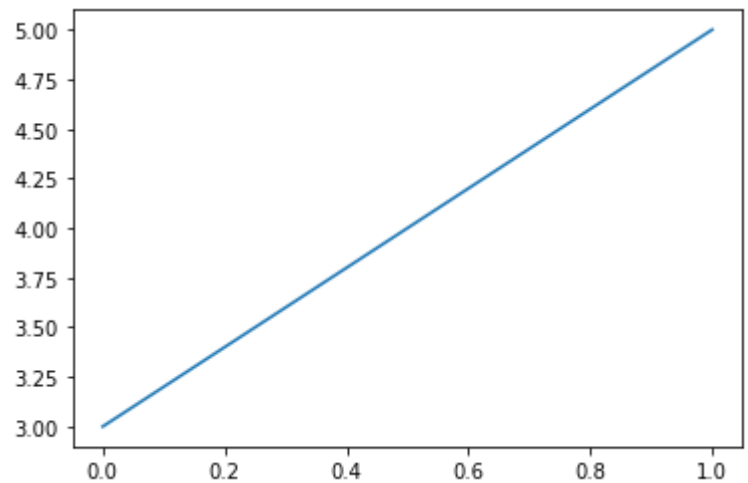
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
```

In [3]:

```
plt.plot([3,5])
```

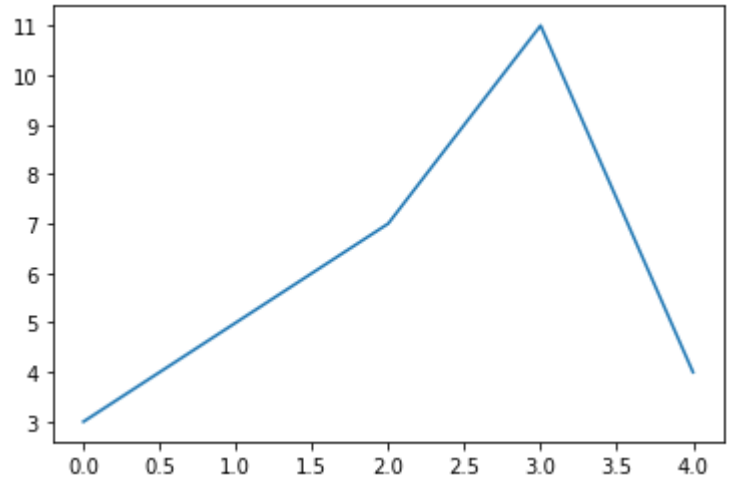
Out[3]:

```
[<matplotlib.lines.Line2D at 0x7f8e40918df0>]
```



In [12]:

```
plt.plot([3, 5, 7,11, 4])
plt.show()
```

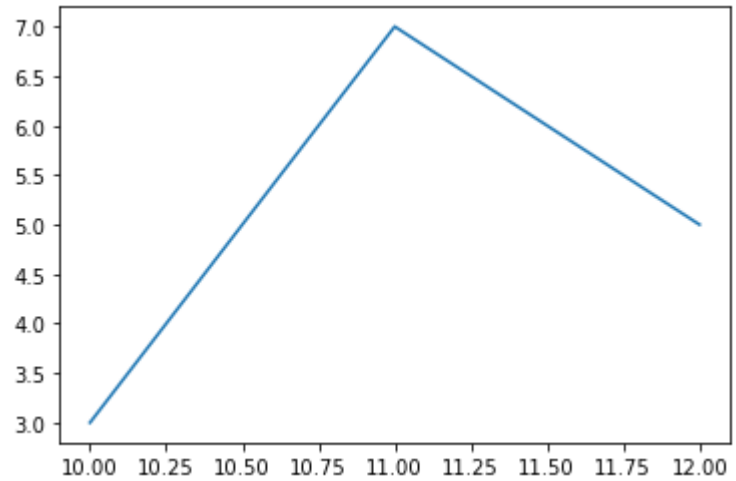


In [17]:

```
plt.plot([10,11,12], [3,7,5])
```

Out[17]:

```
[<matplotlib.lines.Line2D at 0x7f8e3828fb80>]
```



In [ ]:

```

```

In [41]:

```
x = np.arange( 10)
x
```

Out[41]:

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [42]:

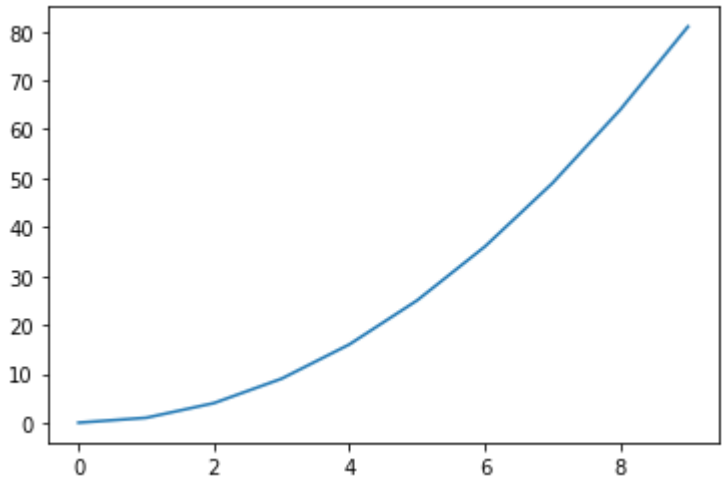
```
# squares
y = x**2
```

In [43]:

```
plt.plot(x, y)
```

Out[43]:

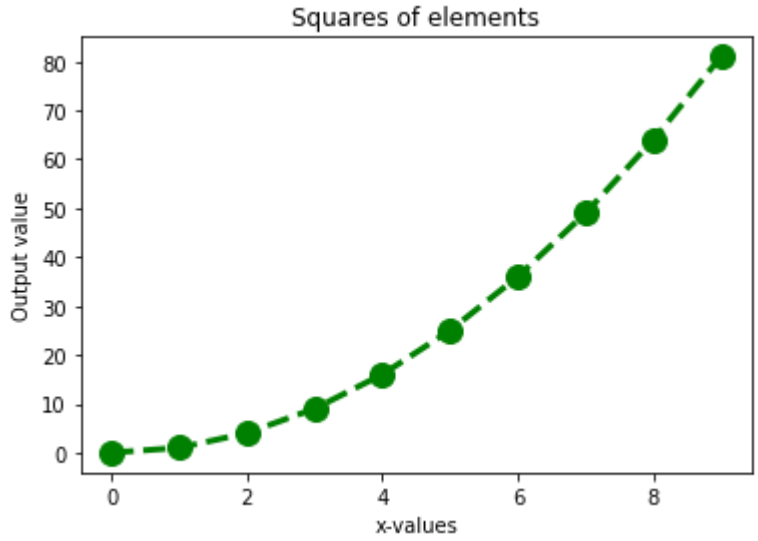
```
[<matplotlib.lines.Line2D at 0x7f8e52c6d610>]
```



Style and Labelling

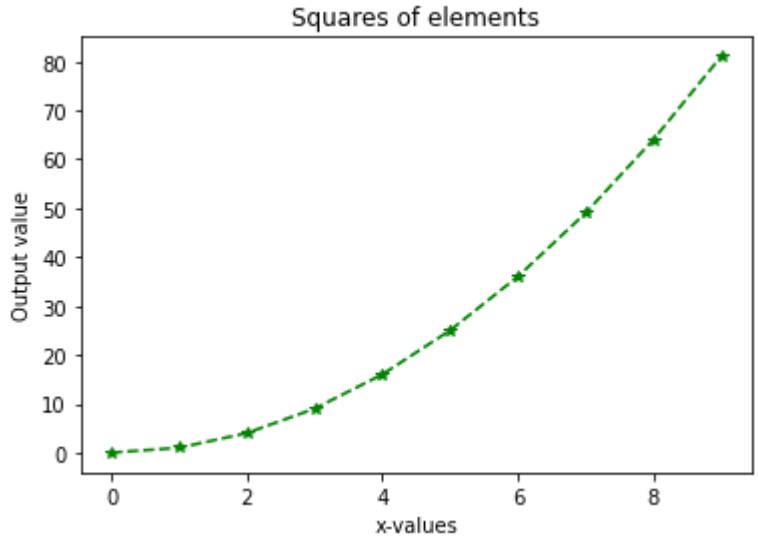
In [60]:

```
x = np.arange( 10)

y = x**2


plt.plot(x, y, color='green', marker='o', markersize=12, linewidth=3, linestyle='das

plt.title("Squares of elements")
plt.xlabel("x-values")
plt.ylabel("Output value")

plt.show()
```



In [62]:

```
x = np.arange( 10)

y = x**2


plt.plot(x, y, 'g*--')

plt.title("Squares of elements")
plt.xlabel("x-values")
plt.ylabel("Output value")

plt.show()
```

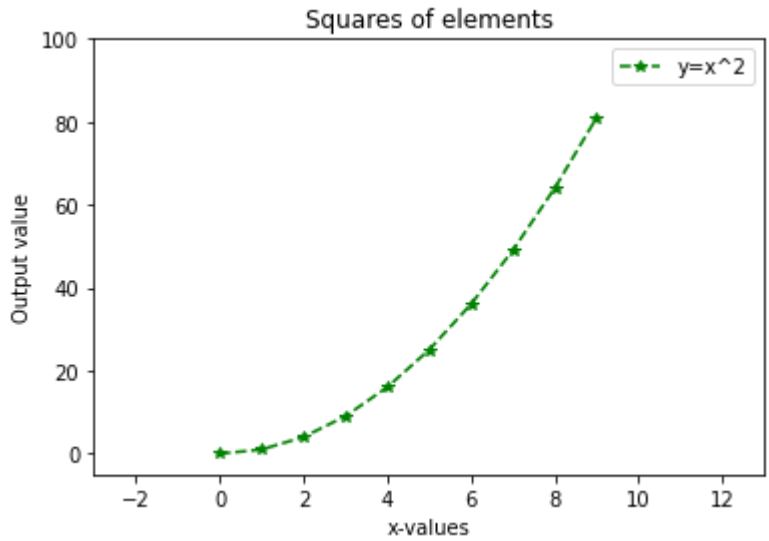


In [ ]:

```

```

In [68]:

```
x = np.arange( 10)

y = x**2


plt.plot(x, y, 'g*--', label="y=x^2")
plt.xlim(-3,  13)
plt.ylim(-5, 100)


plt.legend()

plt.title("Squares of elements")
plt.xlabel("x-values")
plt.ylabel("Output value")

plt.show()
```

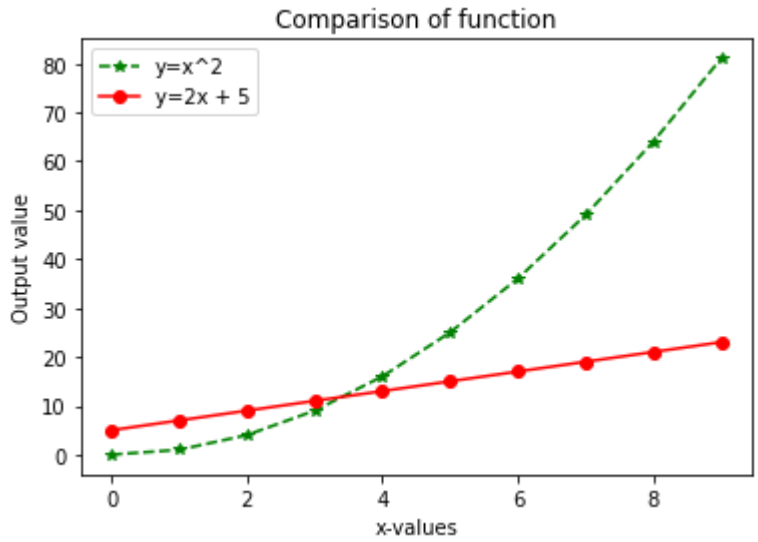


In [ ]:

```

```

In [75]:

```
x = np.arange(10)

y1 = x**2
y2 = 2*x + 5

plt.plot(x, y1, 'g*--', label="y=x^2")
plt.plot(x, y2, 'ro-',label="y=2x + 5")

plt.legend()

plt.title("Comparison of function")
plt.xlabel("x-values")
plt.ylabel("Output value")

plt.show()
```

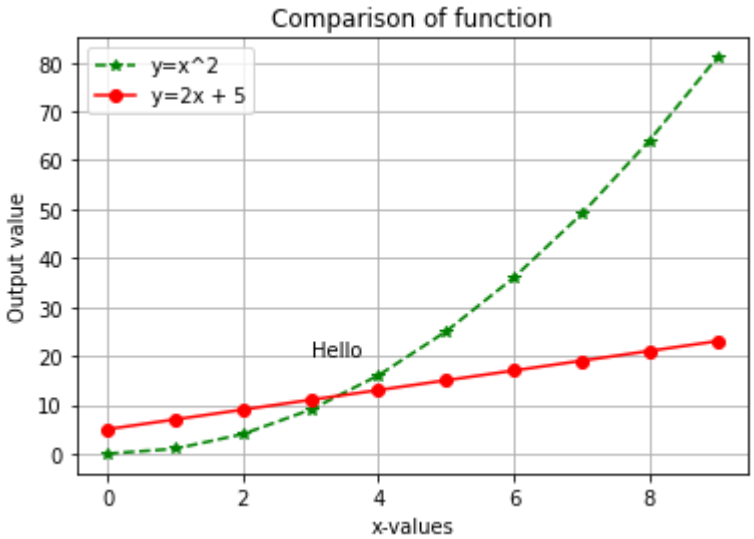


In [ ]:

```
```

In [262]:

```
x = np.arange(10)

y1 = x**2
y2 = 2*x + 5

plt.plot(x, y1, 'g*--', label="y=x^2")
plt.plot(x, y2, 'ro-',label="y=2x + 5")
plt.text(3,20, "Hello")

plt.legend()

# plt.annotate()

plt.grid()

plt.title("Comparison of function")
plt.xlabel("x-values")
plt.ylabel("Output value")

plt.savefig("comparison.png")
plt.show()
```



In [ ]:

```

```

Log function

In [89]:

```
x = np.arange(1, 50)
x[:5]
```

Out[89]:

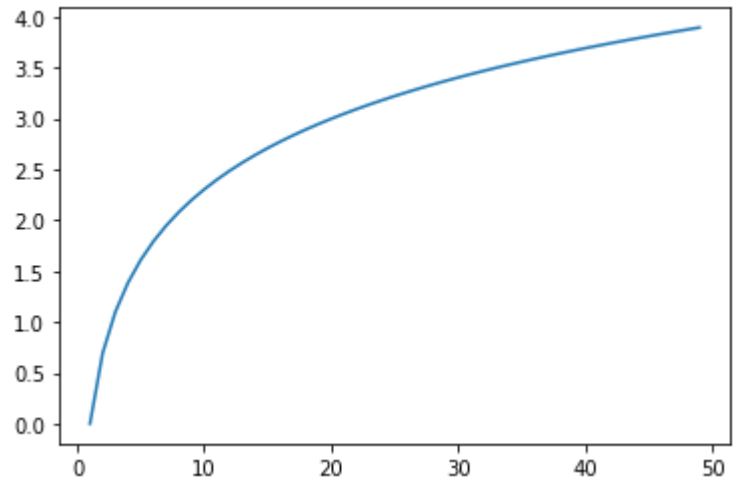
```
array([1, 2, 3, 4, 5])
```

In [90]:

```
y = np.log(x)
```

In [91]:

```
plt.plot(x, y)
plt.show()
```



In [96]:

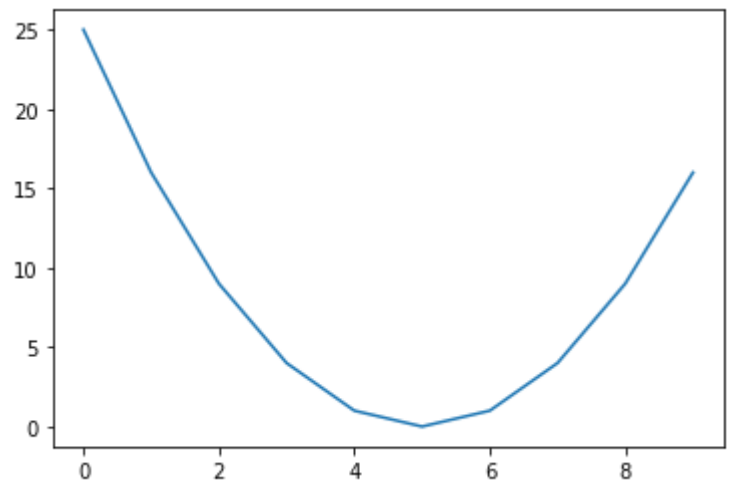
```
x = np.arange(10)
y = [(i-5)**2 for i in x]
```

In [97]:

```
plt.plot(x, y)
```

Out[97]:

```
[<matplotlib.lines.Line2D at 0x7f8e20462b50>]
```



In [ ]:

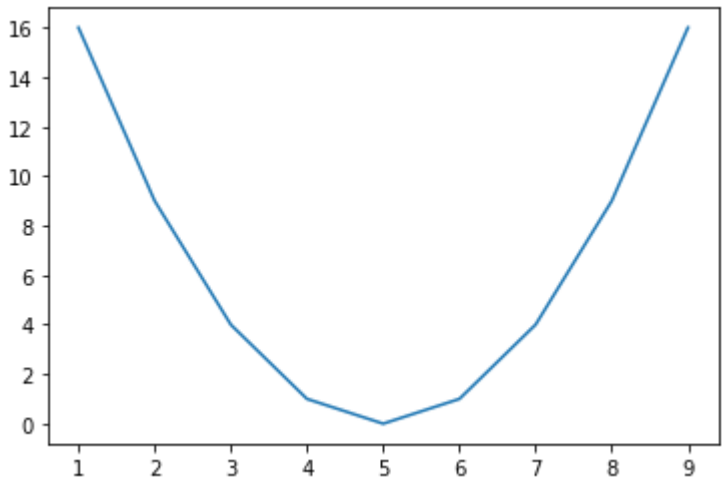
```

```

In [98]:

```
def func(x):
    return (x-5)**2

x = np.arange(1,10)

y1 = np.vectorize(func)(x)

plt.plot(x, y1)
```

Out[98]:

```
[<matplotlib.lines.Line2D at 0x7f8e2030e580>]
```



In [ ]:

```

```

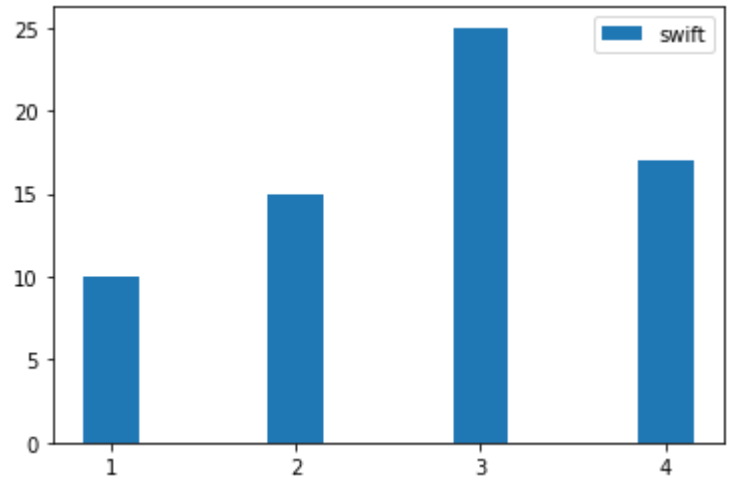
Types of Graph

In [123]:

```
x = np.array([1,2,3,4])
swift_sales = [10, 15, 25, 17]
```

In [114]:

```
plt.bar(x, swift_sales, width=0.3, label="swift")
plt.xticks(x)
plt.legend()
plt.show()
```

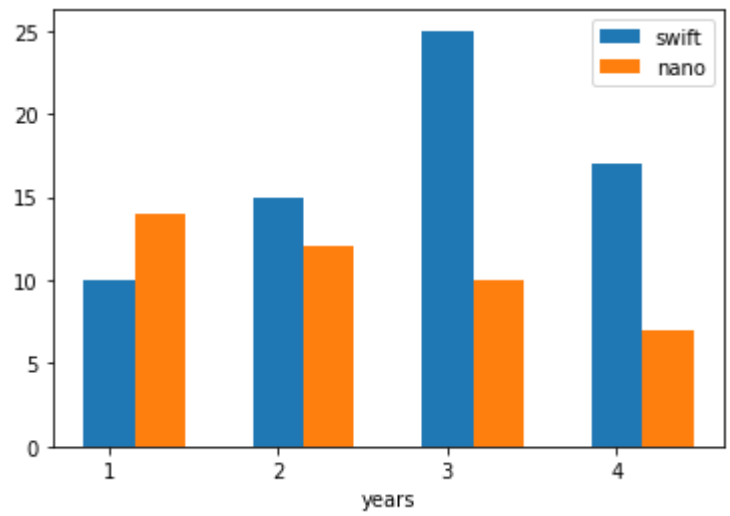


In [116]:

```
nano_sales = [14, 12, 10, 7]
```

In [132]:

```
plt.bar(x, swift_sales, width=0.3, label="swift")
plt.bar(x+0.3, nano_sales, width=0.3, label="nano")

plt.xticks(x)

plt.xlabel("years")
plt.legend()
plt.show()
```



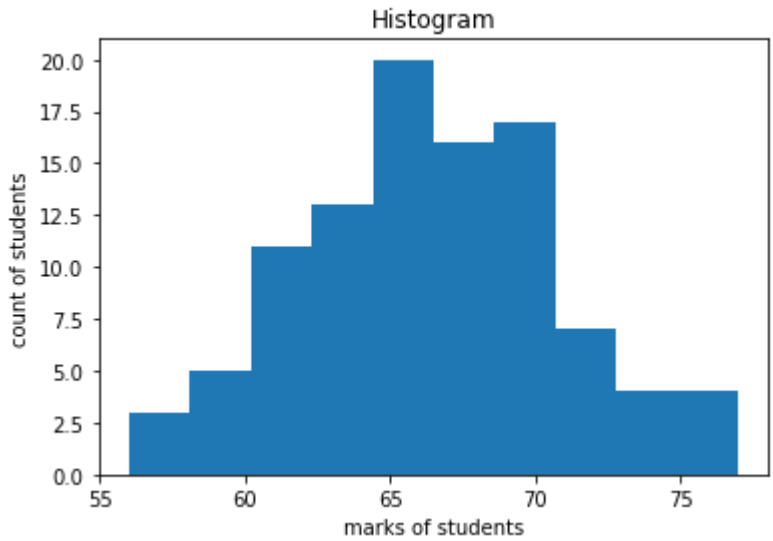
Histogram

In [182]:

```
marks = np.round(np.random.normal(67, 5, size=100))
```

In [183]:

```
count, bins, _ = plt.hist(marks, bins=10)
plt.xlabel("marks of students")
plt.ylabel("count of students")
plt.title("Histogram")
plt.show()
```



In [180]:

```
count
```

Out[180]:

```
array([ 2.,  3.,  6., 15., 18., 18., 16., 14.,  6.,  2.])
```

In [181]:

```
bins
```

Out[181]:

```
array([32. , 38.7, 45.4, 52.1, 58.8, 65.5, 72.2, 78.9, 85.6, 92.3, 99.
])
```

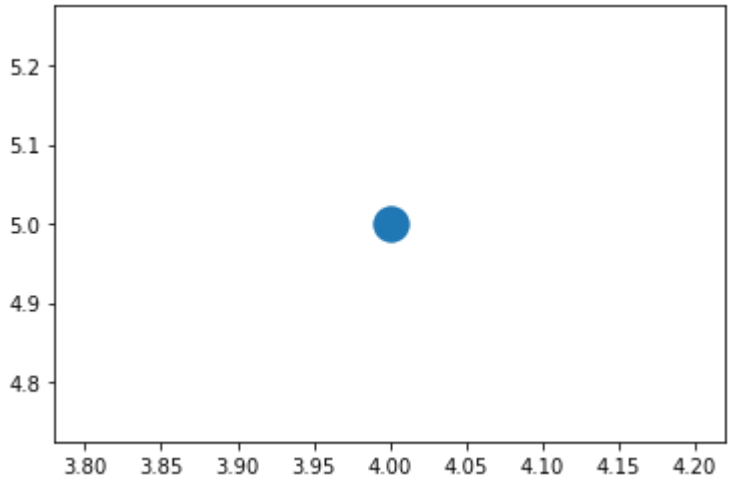
In [ ]:

```

```

ScatterPlot

In [192]:

```
plt.scatter(4,5, s=300)
plt.show()
```

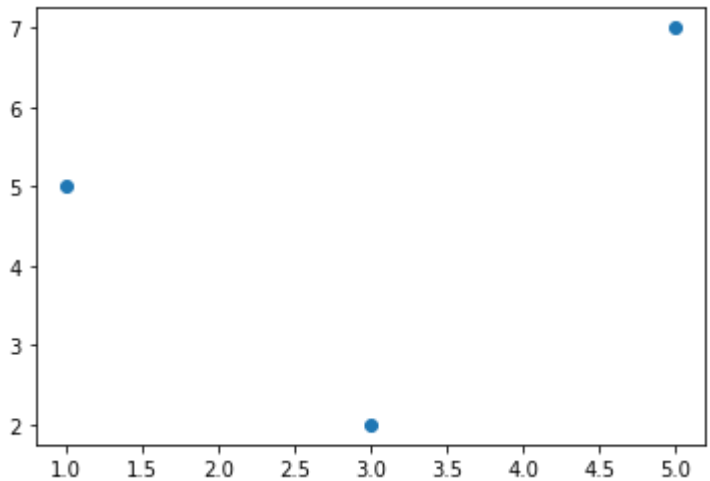


In [194]:

```
x1 = np.random.randn(100)
y1 = np.random.randn(100)
```

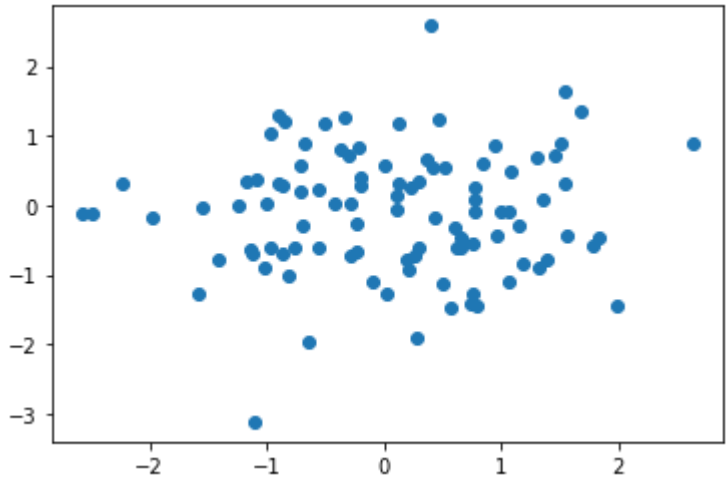
In [196]:

```
plt.scatter([1,5,3], [5, 7,2])
plt.show()
```



In [199]:

```
plt.scatter(x1,y1)
plt.show()
```

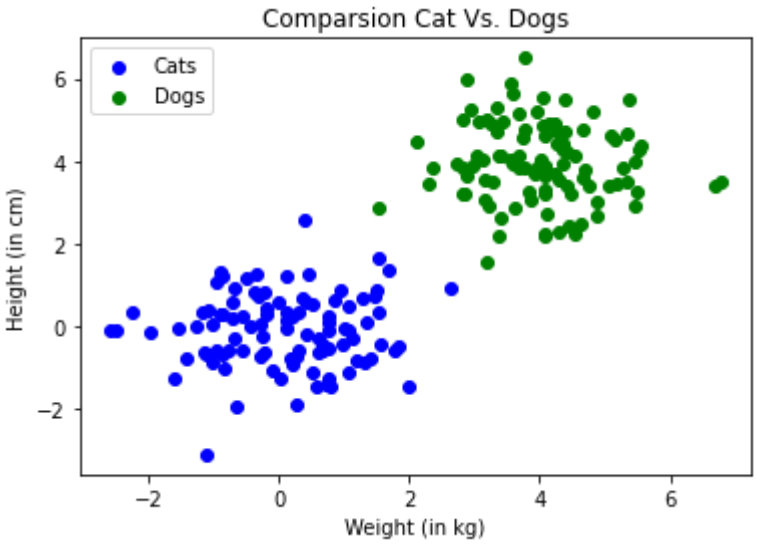


In [202]:

```
x2 = np.random.randn(100) + 4
y2 = np.random.randn(100) + 4
```

In [208]:

```
plt.scatter(x1, y1, label="Cats", color="blue")
plt.scatter(x2, y2, label="Dogs", color="green")

plt.legend()

plt.xlabel("Weight (in kg)")
plt.ylabel("Height (in cm)")
plt.title("Comparsion Cat Vs. Dogs")

plt.show()
```



In [ ]:

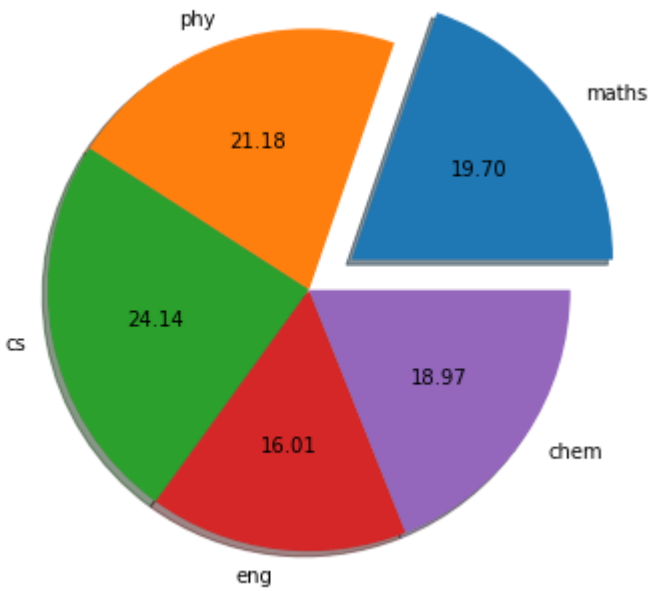
```
```

In [238]:

```
marks = [80, 86, 98, 65, 77]
subjects= ['maths', 'phy', 'cs', 'eng', 'chem']
```

In [246]:

```
plt.figure(figsize=(6, 6))

plt.pie(x= marks, labels=subjects, shadow=True, explode= [0.2,0,0,0,0], autopct =  "
plt.show()
```



In [ ]:

```
```

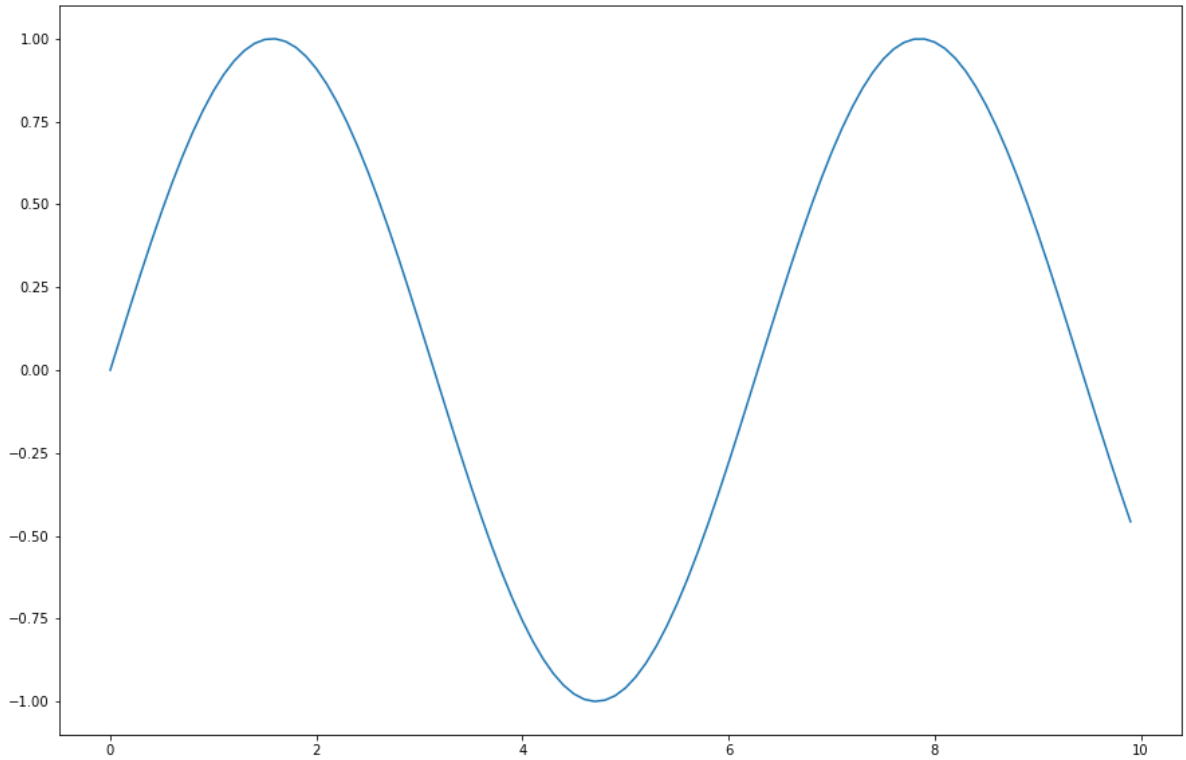
Subplots

In [248]:

```
plt.figure(figsize=(15, 10))

x = np.arange(0, 10, 0.1)
y = np.sin(x)

plt.plot(x, y)
```

Out[248]:

```
[<matplotlib.lines.Line2D at 0x7f8e3893c340>]
```

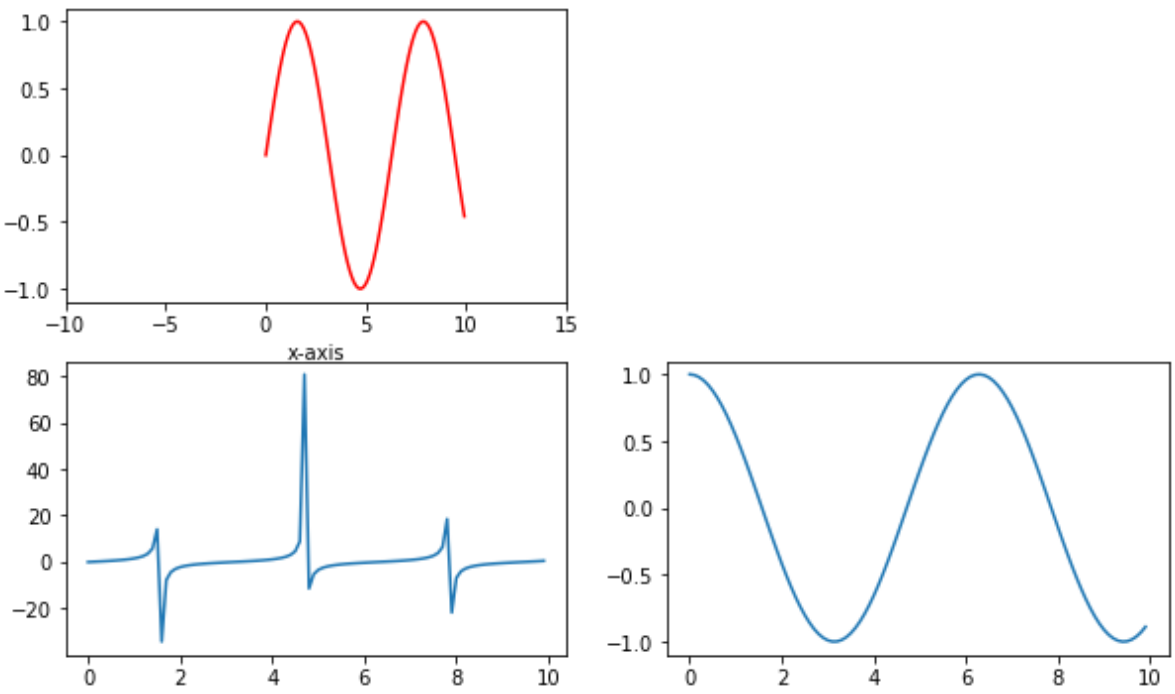


In [ ]:

```
```

In [256]:

```
x = np.arange(0, 10, 0.1)
y = np.sin(x)


plt.figure(figsize=(10, 6))


plt.subplot(2, 2, 1)
plt.plot(x, y, 'r')
plt.xlabel("x-axis")
plt.xlim(-10, 15)



plt.subplot(2,2,3)
plt.plot(x, np.tan(x))


plt.subplot(2,2,4)
plt.plot(x, np.cos(x))




plt.show()
```



In [ ]:

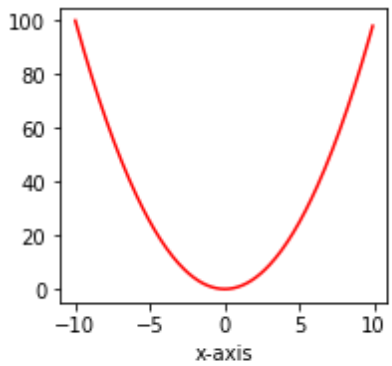
```

```

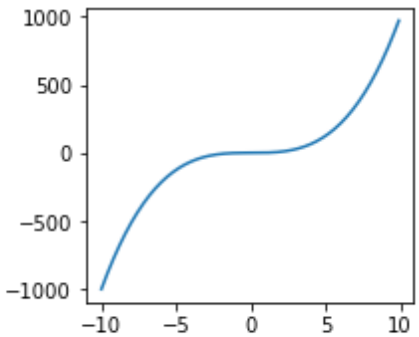
In [ ]:

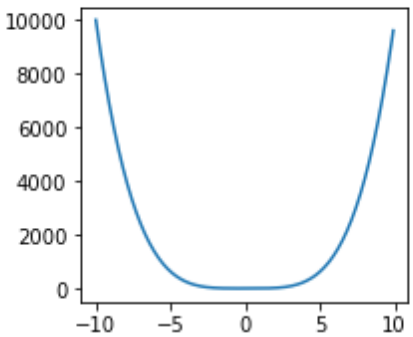
```

```

In [264]:

```
x = np.arange(-10, 10, 0.1)


plt.figure(figsize=(10, 6))


plt.subplot(2, 3, 1)
plt.plot(x, x**2, 'r')
plt.xlabel("x-axis")
# plt.xlim(-10, 15)
# plt.axis("off")


plt.subplot(2,3,3)
plt.plot(x, x**3)


plt.subplot(2,3,5)
plt.plot(x, x**4)


plt.show()
```



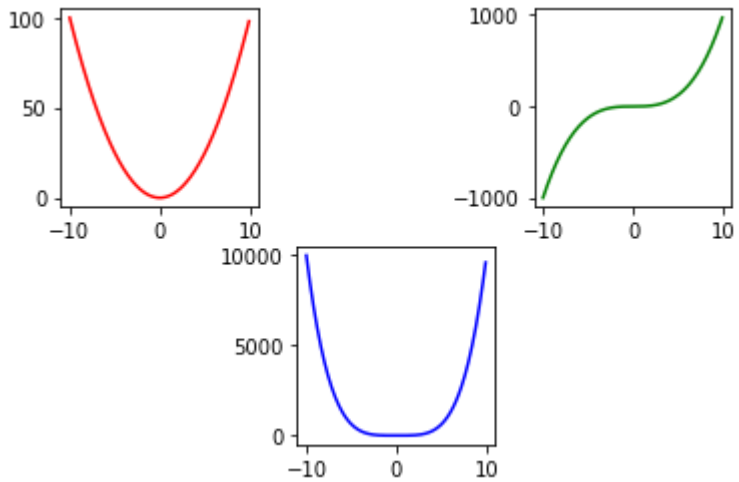




In [ ]:

In [278]:

```
fig = plt.figure()

ax = fig.add_subplot(2,3,1)
ax.plot(x, x**2, 'r')

ax = fig.add_subplot(2,3,3)
ax.plot(x, x**3, 'g')


ax = fig.add_subplot(2,3,5)
ax.plot(x, x**4, 'b')


plt.savefig()
plt.show()
```



In [ ]:

```

```

3D Plots

In [279]:

```
a = np.array([0,1,2,3])
b = np.array([0,1,2])
```

In [281]:

```
a,b = np.meshgrid(a, b)
```

In [282]:

```
a
```

Out[282]:

```
array([[0, 1, 2, 3],
       [0, 1, 2, 3],
       [0, 1, 2, 3]])
```

In [283]:

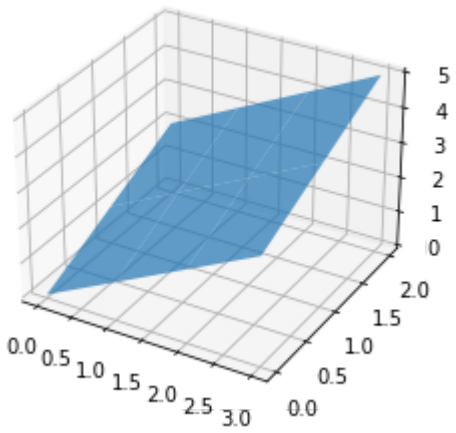
```
b
```

Out[283]:

```
array([[0, 0, 0, 0],
       [1, 1, 1, 1],
       [2, 2, 2, 2]])
```

In [293]:

```
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.plot_surface(a, b, a+b, alpha=0.7)
plt.show()
```



In [297]:

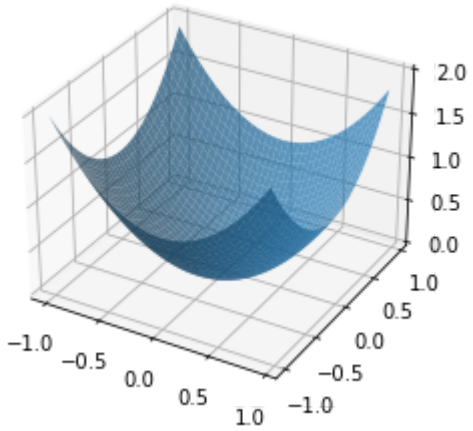
```
a = np.arange(-1, 1, 0.05)
```

In [298]:

```
a,b = np.meshgrid(a, a)
```

In [299]:

```
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.plot_surface(a, b, a**2 + b**2, alpha=0.7)
plt.show()
```

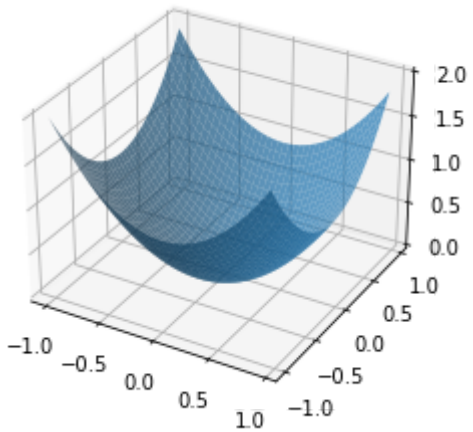


In [302]:

```
# in jupyter notebook
# %matplotlib inline


# plot outside jupyter
# %matplotlib
```

In [303]:

```
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.plot_surface(a, b, a**2 + b**2, alpha=0.7)
plt.show()
```



In [ ]:

```

```