MDM_AI-ML_Python_Lec4

January 6, 2025

Data Analytics with Python

Lecture 4 (MDM)

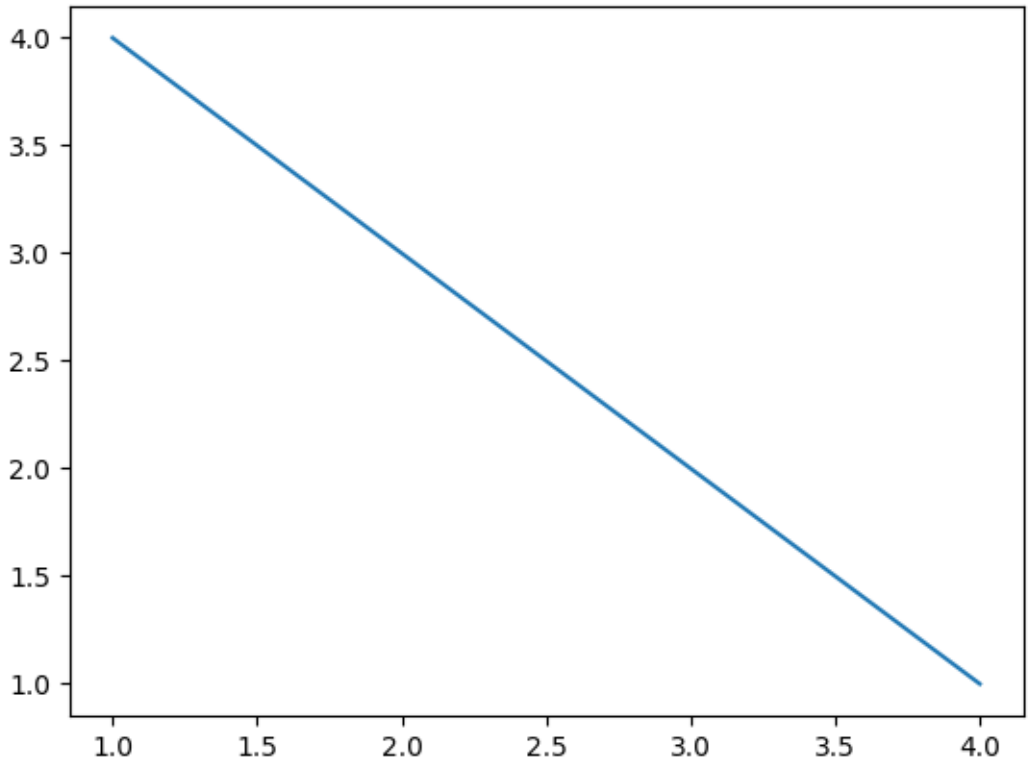
By Ajit Kumar (ICT Mumbai)

Jan. 02, 2025

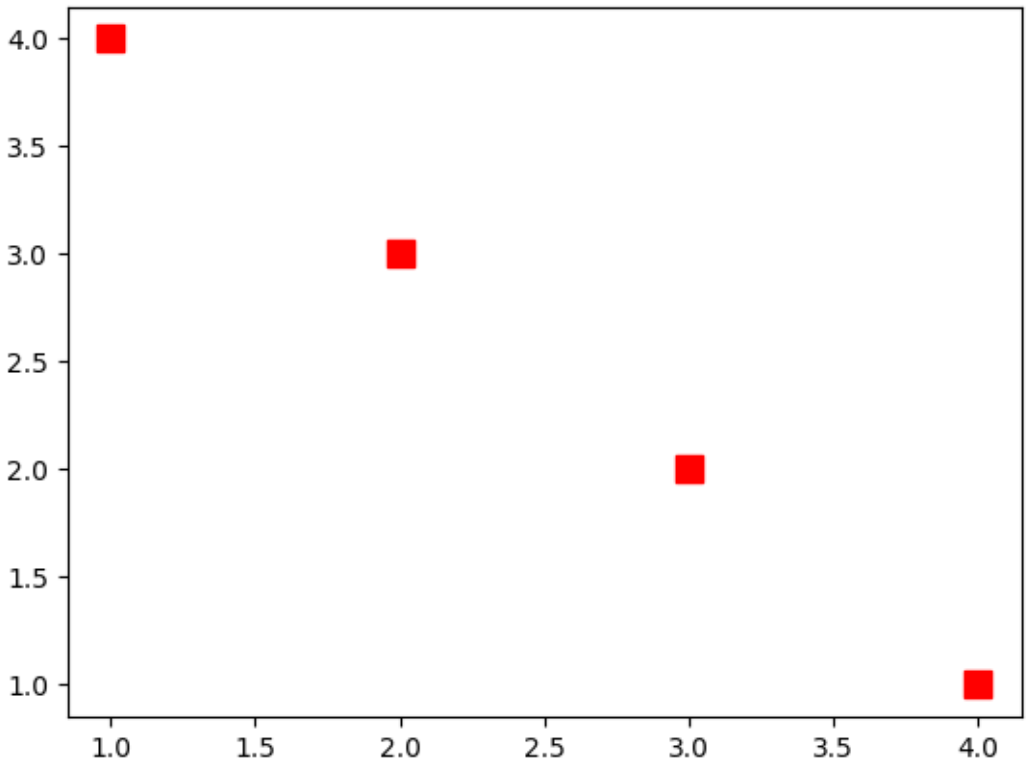
0.1 Matplotlib for plotting

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
# Create a basic plot
plt.plot([1, 2, 3, 4], [4, 3, 2, 1])
plt.show()
```

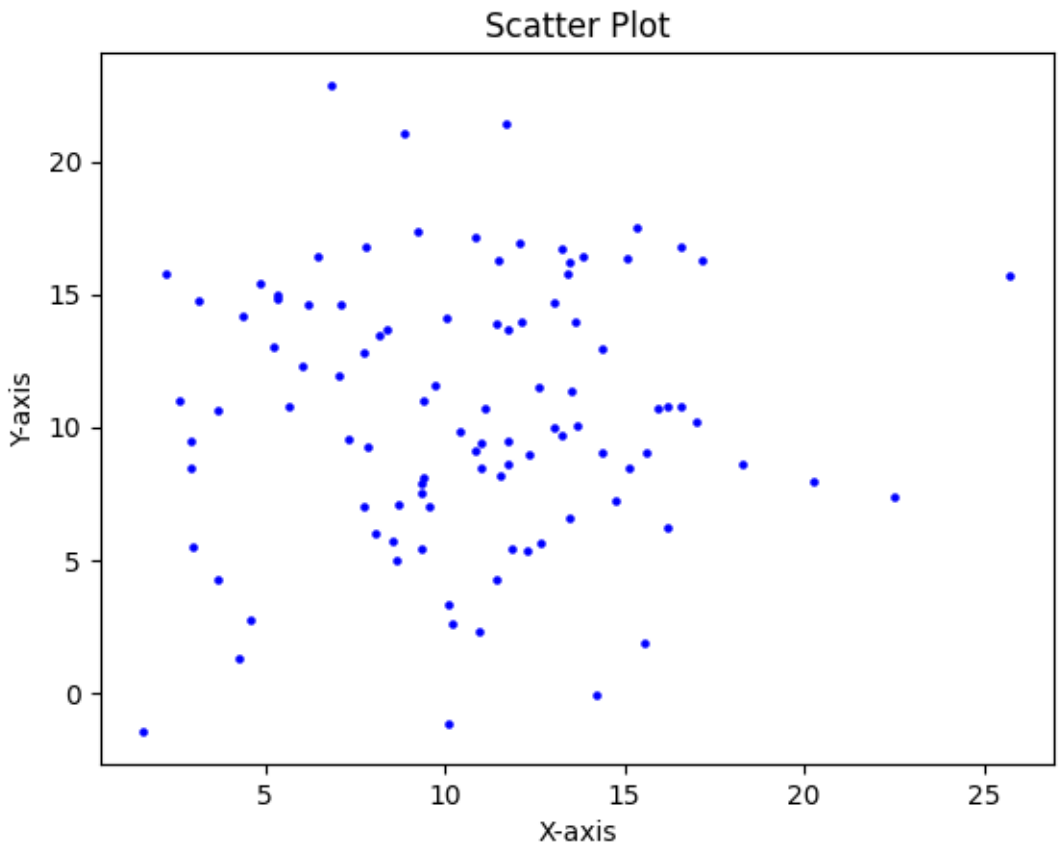


[3]:
```
plt.plot([1, 2, 3, 4], [4, 3, 2, 1], 'rs', markersize=10)
plt.show()
```

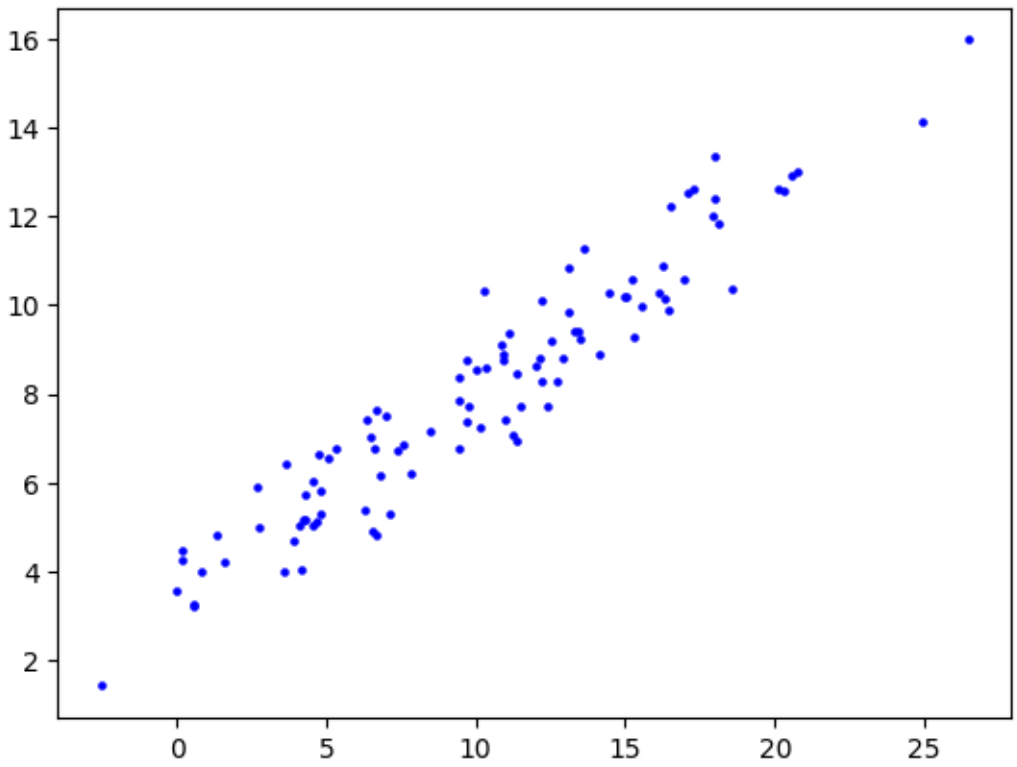


0.1.1 Scatter Plot

[4]:
```
import numpy as np
import matplotlib.pyplot as plt
x = np.random.normal(10,5,100)
y = np.random.normal(10,5,100)
plt.scatter(x,y,c='b',s=5)
plt.title("Scatter Plot")
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```



[5]:
```
import matplotlib.pyplot as plt
import numpy as np
x = np.random.normal(10,5,100)
y = 3 + .5*x + np.random.normal(0,1,100)
vec = np.array([x,y])
plt.scatter(x,y,c='b',s=5)
plt.show()
```



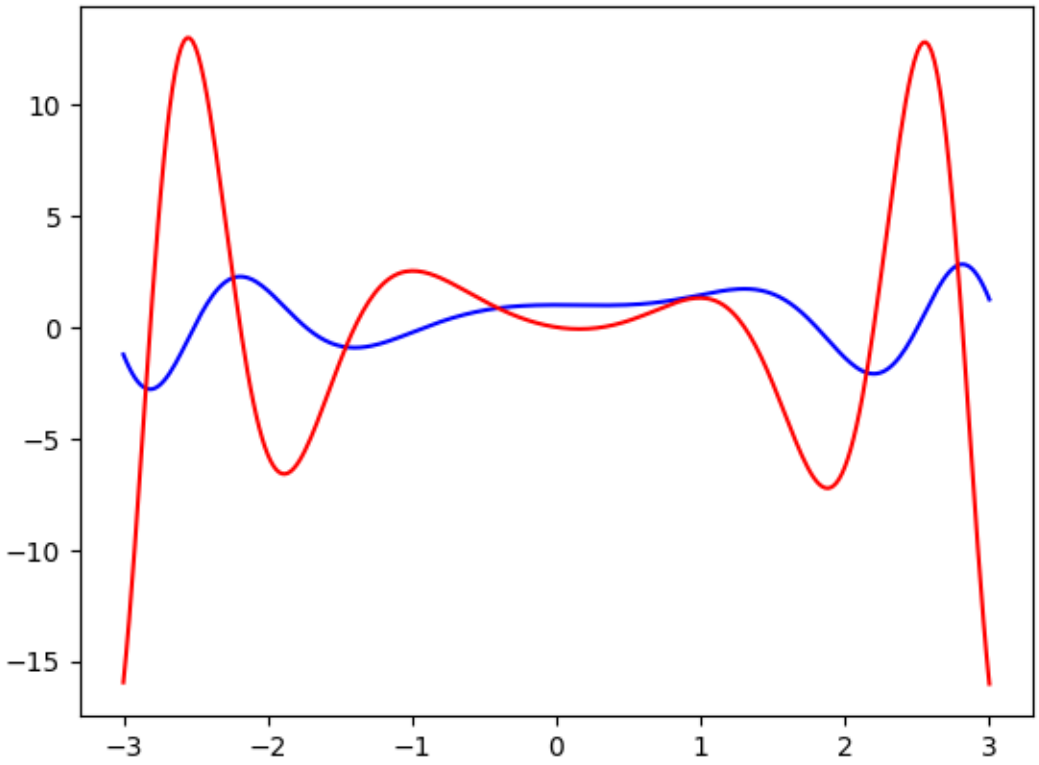
[ ]:

0.1.2 Plotting Curves

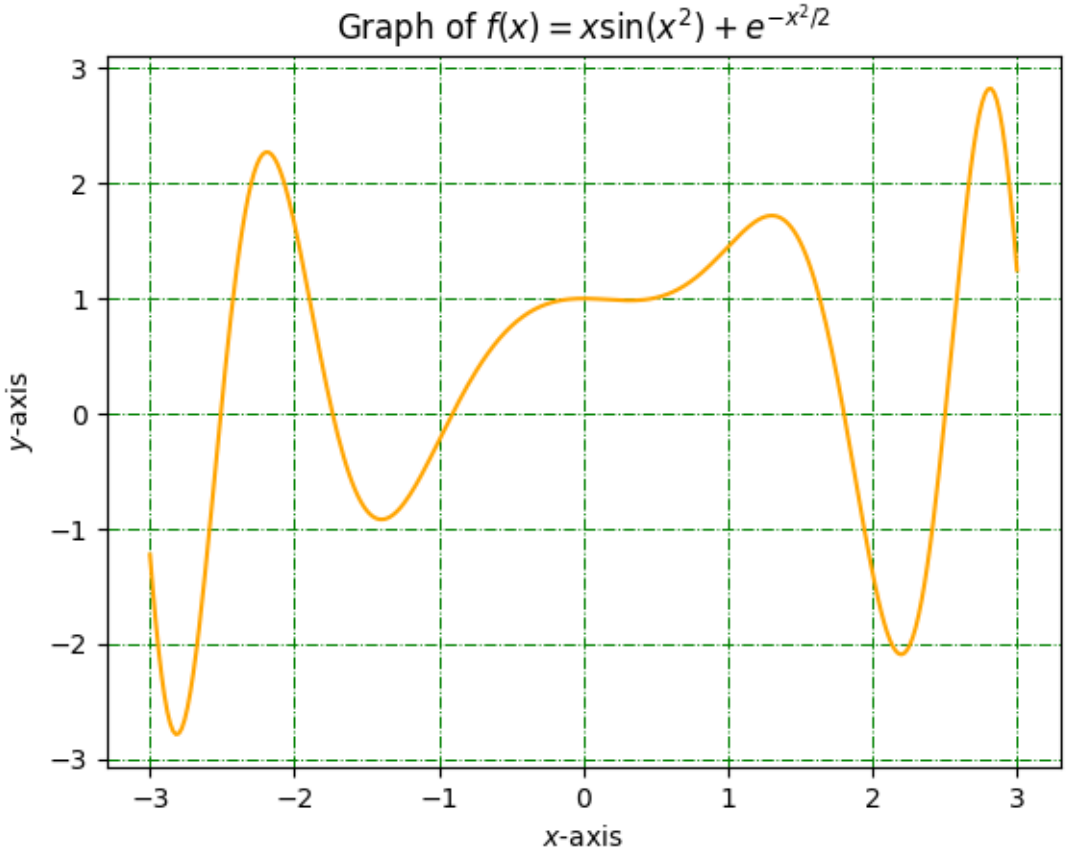
Plotting graph of

$$f(x) = x\sin(x^2) + e^{-x^2/2}.$$

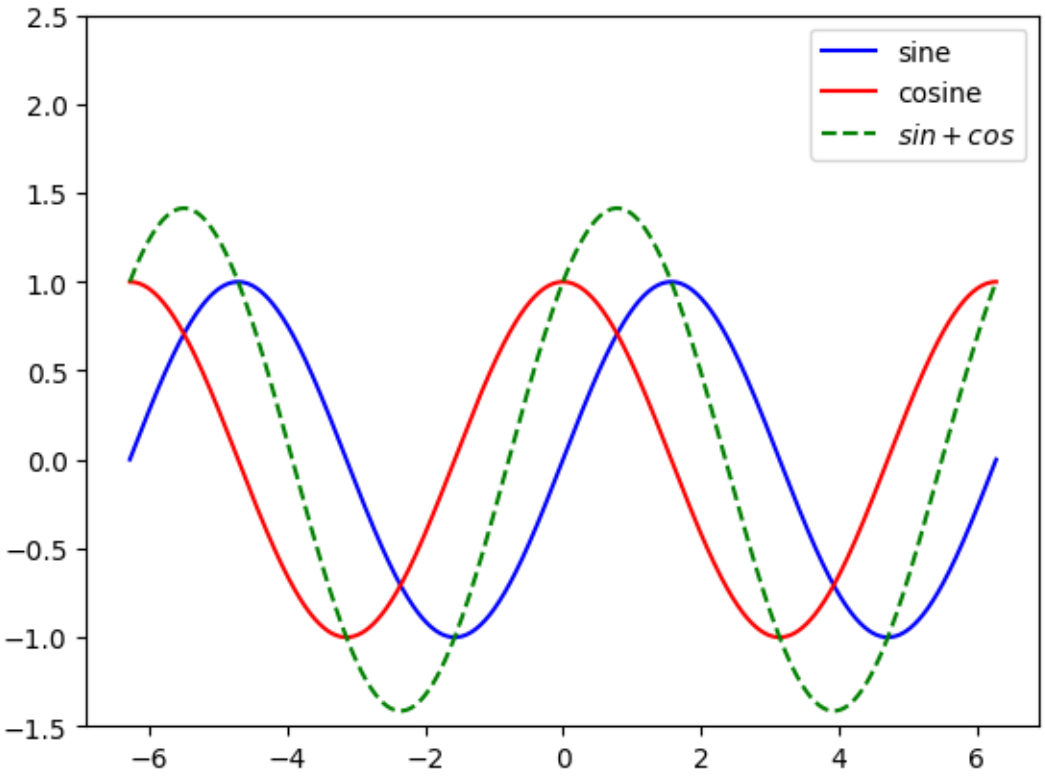
[6]:
```
x = np.linspace(-3,3,301)
y = x*np.sin(x**2)+np.exp(-x**2/2)
y1 = 2*x**2*np.cos(x**2)+np.sin(x**2)-x*np.exp(-x**2/2)
plt.plot(x,y,'b')
plt.plot(x,y1,'r')
plt.show()
```



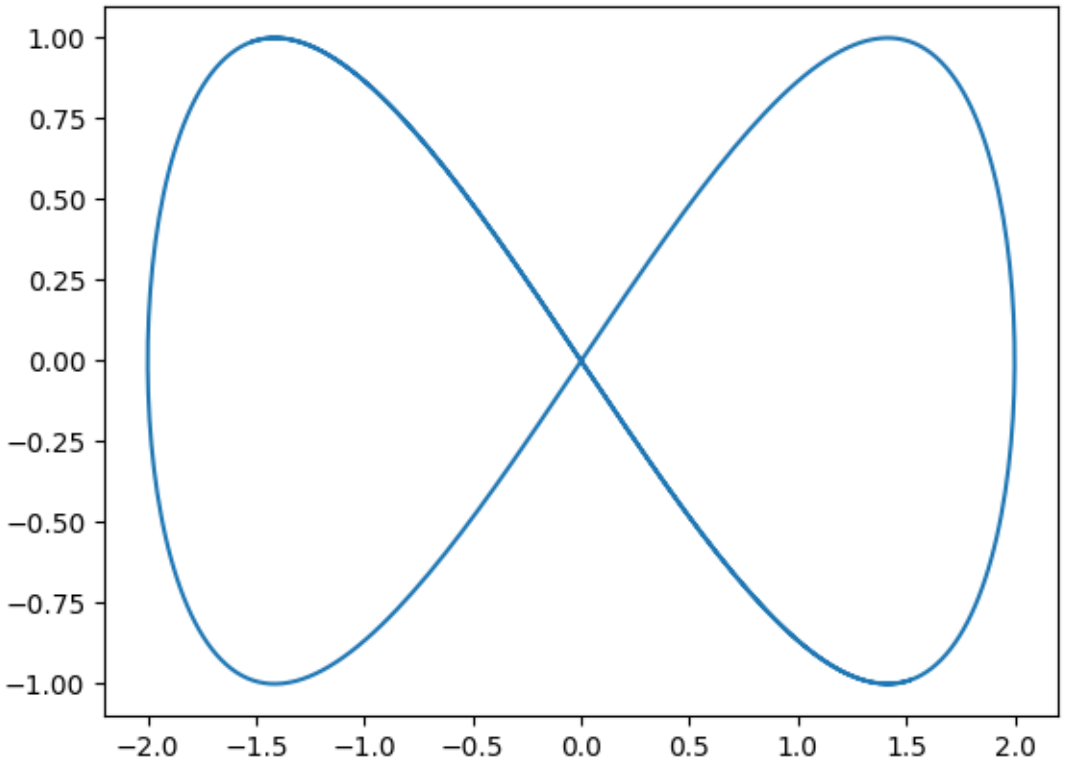
[7]:
```
## Plotting with gridlines and labels
plt.plot(x,y,'orange')
plt.xlabel('$x$-axis')
plt.ylabel('$y$-axis')
plt.title('Graph of $f(x)=x\sin(x^2)+e^{-x^2/2}$')
plt.grid(color='green', linestyle='-.', linewidth=0.7)
plt.show()
```



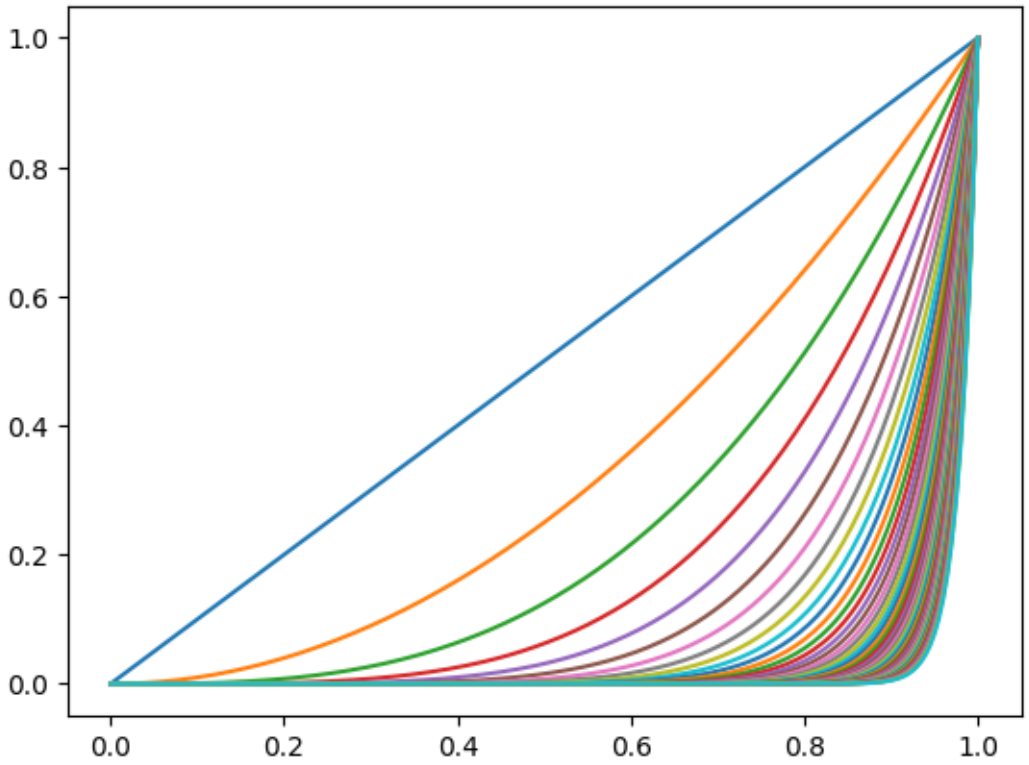
[8]:
```
## Adding legends
import numpy as np
import pylab
x = np.linspace(-2*np.pi, 2*np.pi, 1000)
y1 = np.sin(x)
y2 = np.cos(x)
y3 = y1+y2
pylab.plot(x, y1, '-b', label='sine')
pylab.plot(x, y2, '-r', label='cosine')
pylab.plot(x, y3, '--g', label='$sin+cos$')
pylab.legend()
pylab.ylim(-1.5, 2.5)
pylab.show()
```



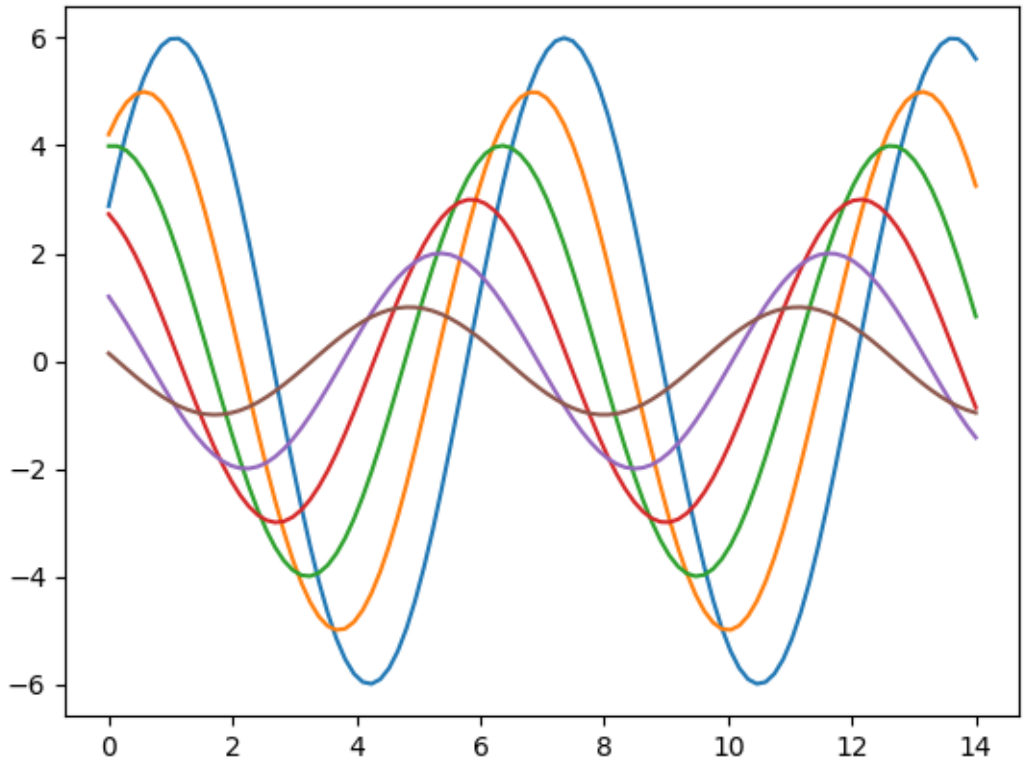
[9]:
```
## Figure infinity in parametric form
from matplotlib import pyplot as plt
import numpy as np
t = np.linspace(-4,4,401)
a = 2
x = a*np.sin(t)
y = a*np.sin(t)*np.cos(t)
plt.plot(x,y)
plt.show()
```



[10]:
```
## Plotting x^i for i in 1 to 50
x = np.linspace(0, 1,201)
for i in range(1,51):
    plt.plot(x, x**i)
plt.show()
```

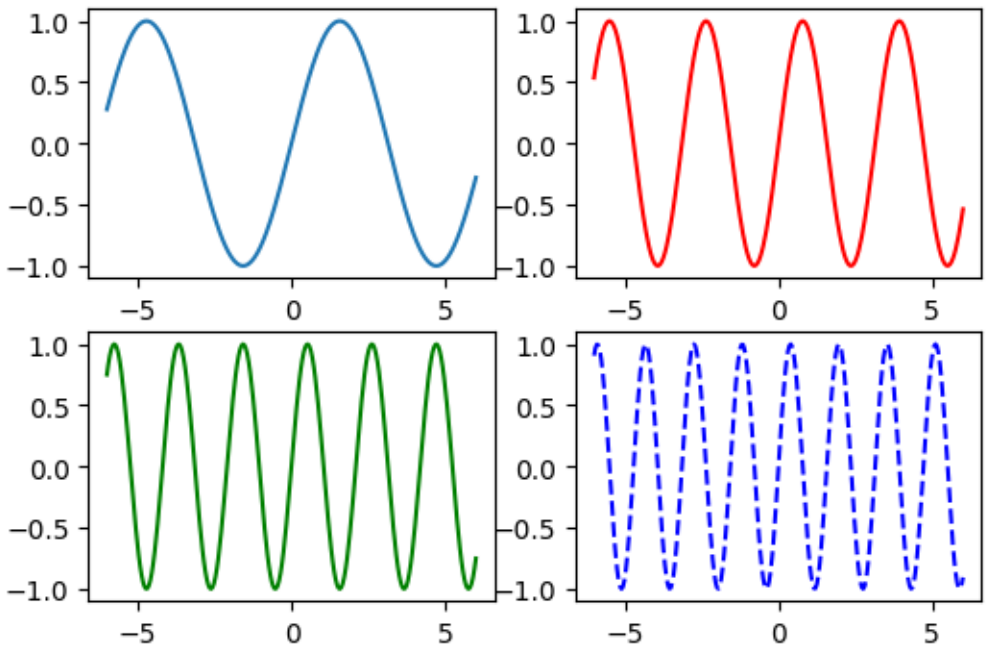


[11]:
```
## Plotting Sinusoidal function
def sinplot():
    fig, ax = plt.subplots()
    x = np.linspace(0, 14, 100)
    for i in range(1, 7):
        ax.plot(x, np.sin(x + i * .5) * (7 - i))
    return ax
ax = sinplot()
```



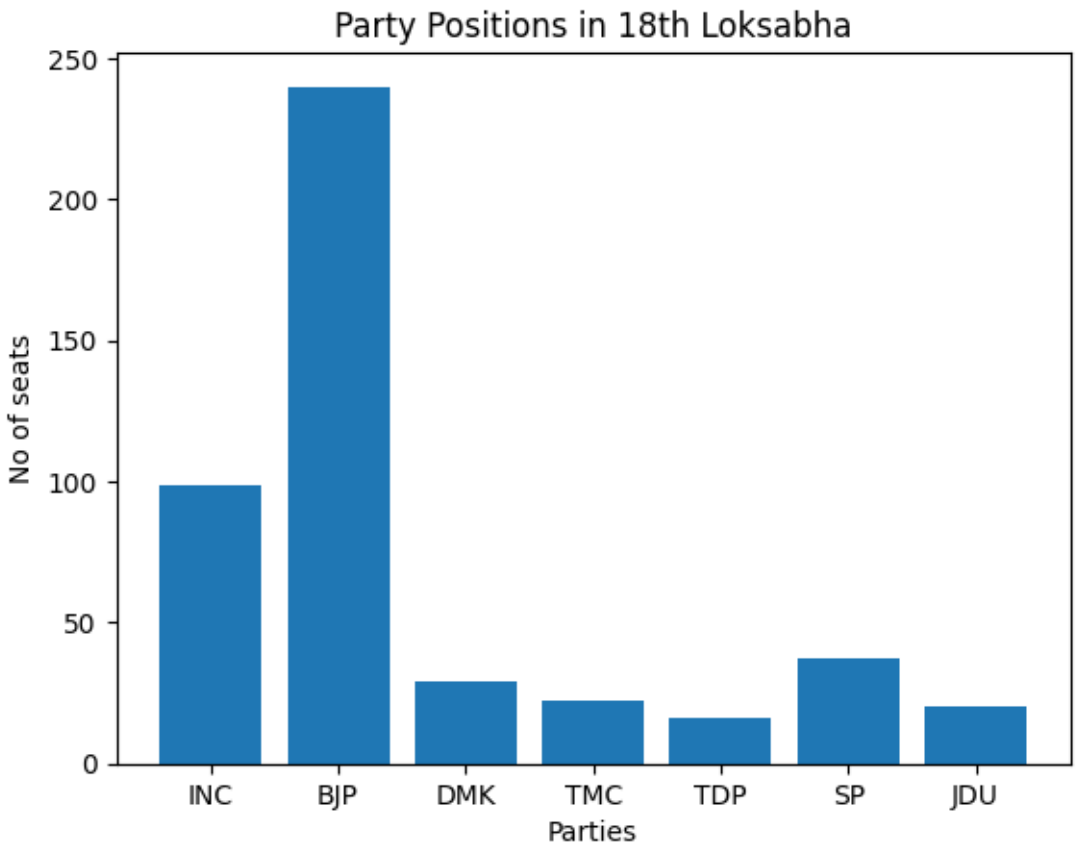
0.1.3 Plotting side by side

[12]:
```
import matplotlib.pyplot as plt
import numpy as np
t = np.linspace(-6, 6, 256)
plt.figure(figsize=(6, 4))
# First graph
plt.subplot(2, 2, 1)
plt.plot(t,np.sin(t))
# Second graph
plt.subplot(2, 2, 2)
plt.plot(t,np.sin(2*t),"r")
# Third graph
plt.subplot(2, 2, 3)
plt.plot(t,np.sin(3*t),"g")
# Fourth graph
plt.subplot(2, 2, 4)
plt.plot(t,np.sin(4*t),"--b")
plt.show()
```

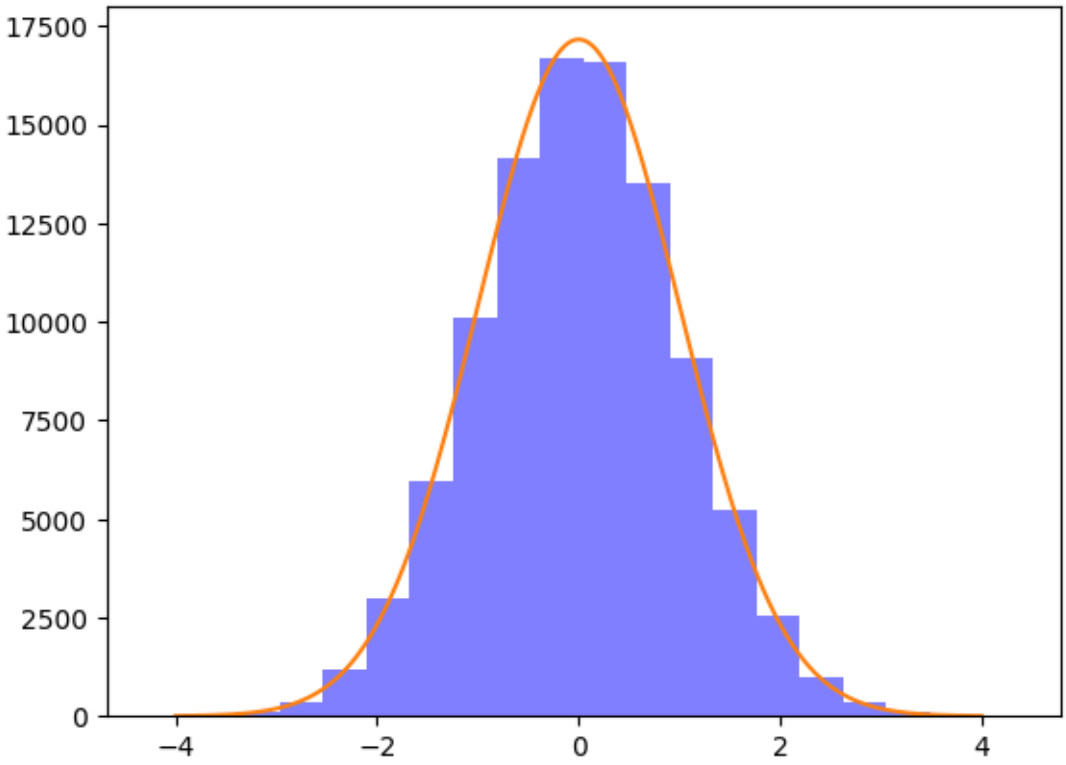


0.1.4 Statistical Plots

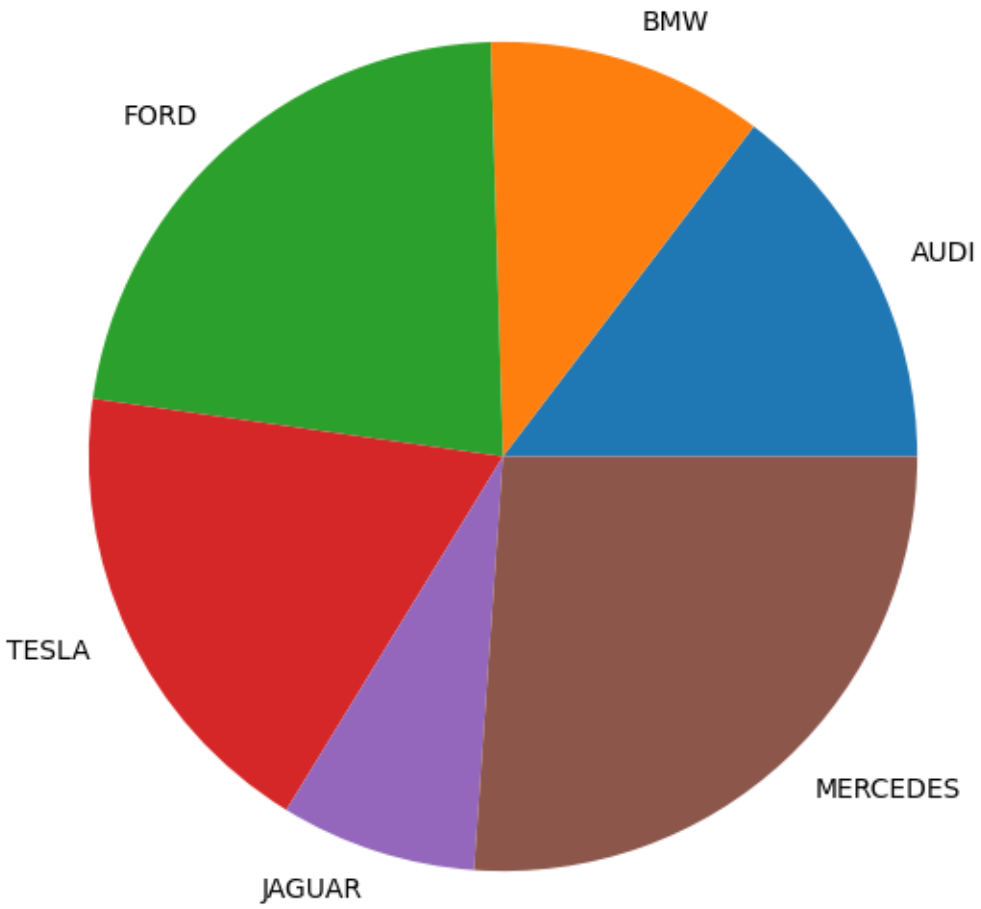
[13]:
```
## Bar plot of Political Parties in 18th Loksabha Gen. Election
party = ['INC','BJP','DMK','TMC','TDP','SP','JDU']
seats = [99,240,29,22,16,37,20]
plt.xlabel('Parties')
plt.ylabel('No of seats')
plt.title('Party Positions in 18th Loksabha')
plt.bar(party,seats)
plt.show()
```



[14]:
```
## Histograms and Density plot
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
N = 100000
x = np.random.randn(N)
num_bins = 20
plt.hist(x, bins=num_bins, facecolor='blue', alpha=0.5)
y = np.linspace(-4, 4, 1000)
bin_width = (x.max() - x.min()) / num_bins
plt.plot(y, stats.norm.pdf(y) * N * bin_width)
plt.show()
```

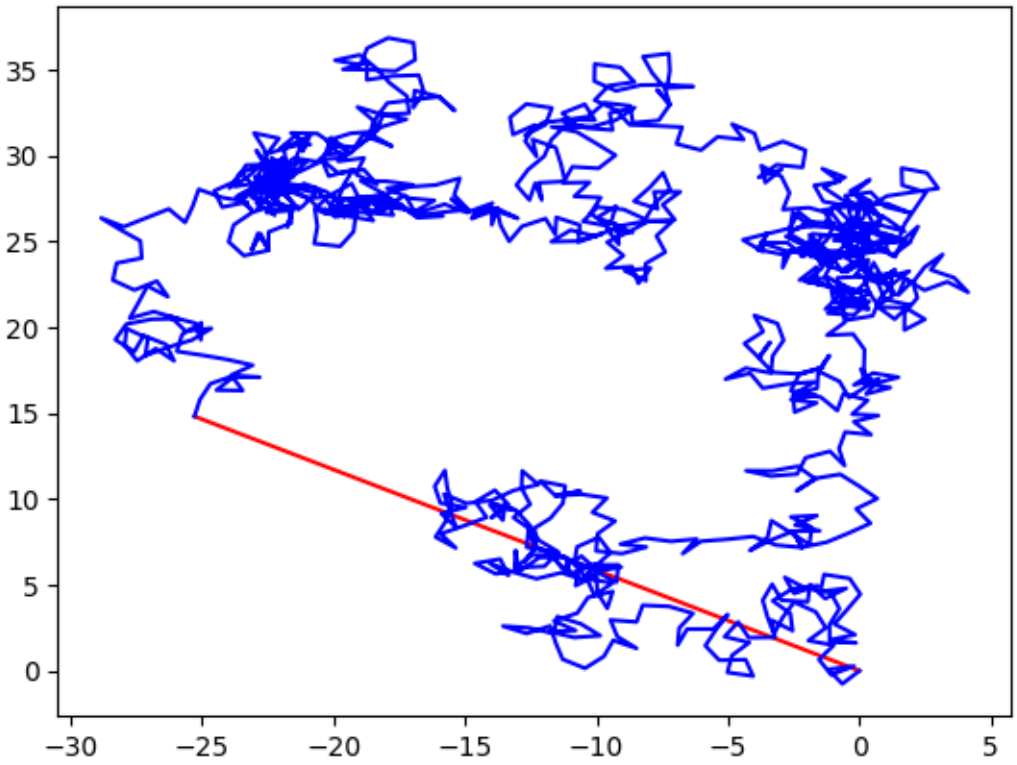


[15]:
```
## Pichart plot
from matplotlib import pyplot as plt
import numpy as np
cars = ['AUDI', 'BMW', 'FORD',
'TESLA', 'JAGUAR', 'MERCEDES']
sold = [23, 17, 35, 29, 12, 41]
fig = plt.figure(figsize =(10, 7))
plt.pie(sold, labels = cars)
plt.show()
```



0.1.5 Plotting a random walk

[16]:
```
import numpy as np
import matplotlib.pyplot as plt
n, l, x, y = 1000, 1, 0, 0
p = [[0,0]]

for k in range(n):
    theta =  2*np.pi * np.random.random()
    x, y = x + l * np.cos(theta), y + l * np.sin(theta)
    p.append([x, y])
plt.plot([p[0][0],p[n][0]],[p[0][1],p[n][1]],'r')
plt.plot([p[i][0] for i in range(n+1)],[p[i][1] for i in range(n+1)],'b')
plt.show()
```



0.2 SciPy Module

[17]:
```
import scipy as sp
```

[18]:
```
# help(sp)
```

0.2.1 Basic Stats using SciPy

[19]:
```
import numpy as np
from scipy import stats
data = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
x = np.array([1,2,3,4,5,6,7,8,9])
print( x.max(),x.min(),x.mean(),x.var())
# Calculate mean and standard deviation
mean_val = np.mean(data)
std_dev = np.std(data)
# Perform basic statistical tests
t_stat, p_value = stats.ttest_1samp(data, popmean=5)
print("t_stat:" , t_stat)
print("p_value:", p_value)
```

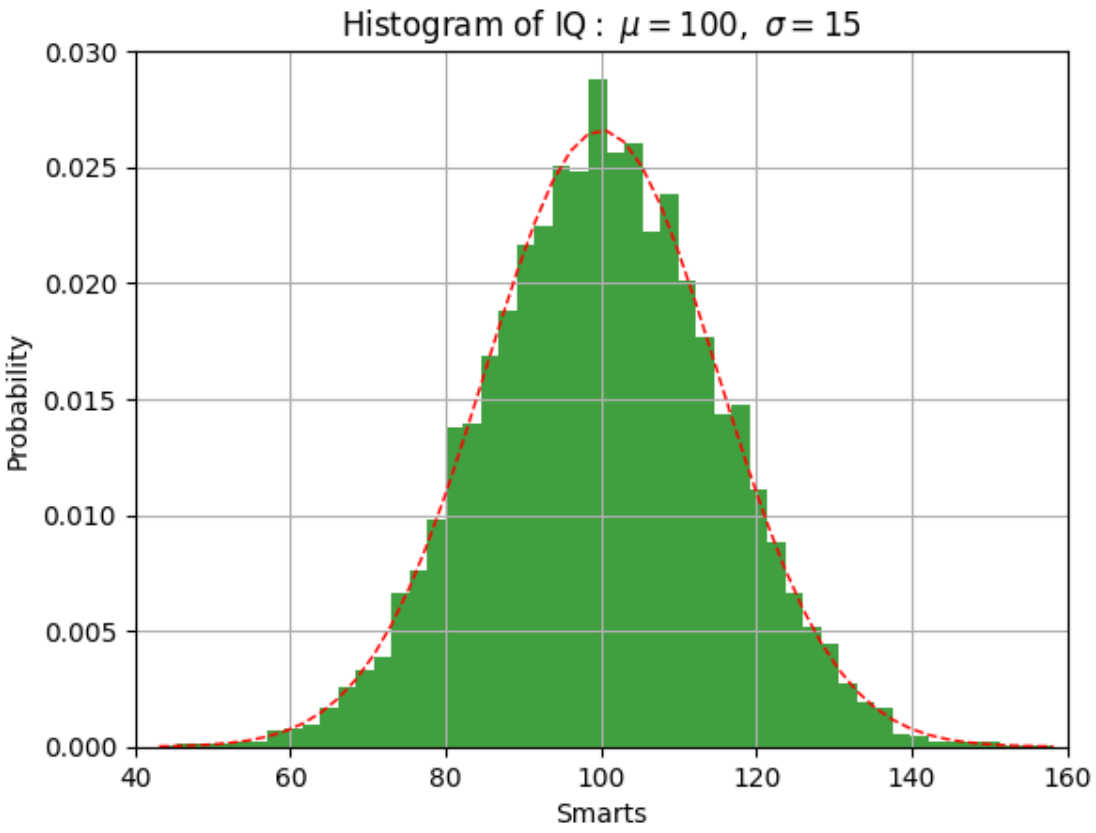
```
9 1 5.0 6.666666666666667
```

```
t_stat: 0.0
p_value: 1.0
```

[20]:
```
from scipy import stats
rvs1 = stats.norm.rvs(loc = 5,scale = 10,size = 500)
rvs2 = stats.norm.rvs(loc = 5,scale = 10,size = 500)
print (stats.ttest_ind(rvs1,rvs2))
```

```
Ttest_indResult(statistic=0.5207835439518013, pvalue=0.6026330896920176)
```

[21]:
```
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats

# create the data
mu, sigma = 100, 15
x = mu + sigma*np.random.randn(10000)

# create the figure and axes objects
fig, ax = plt.subplots()

# the histogram of the data
n, bins, patches = ax.hist(x, 50, density=1, facecolor='green', alpha=0.75)

# add a 'best fit' line
y = scipy.stats.norm.pdf(bins, mu, sigma)
l = ax.plot(bins, y, 'r--', linewidth=1)

# annotate the plot
ax.set_xlabel('Smarts')
ax.set_ylabel('Probability')
ax.set_title(r'$\mathrm{Histogram\ of\ IQ:}\ \mu=100,\ \sigma=15$')
ax.axis([40, 160, 0, 0.03])
ax.grid(True)
```


Histogram of IQ : $\mu = 100,\ \sigma = 15$
Probability
Smarts
0.030
0.025
0.020
0.015
0.010
0.005
0.000
40
60
80
100
120
140
160

[ ]: