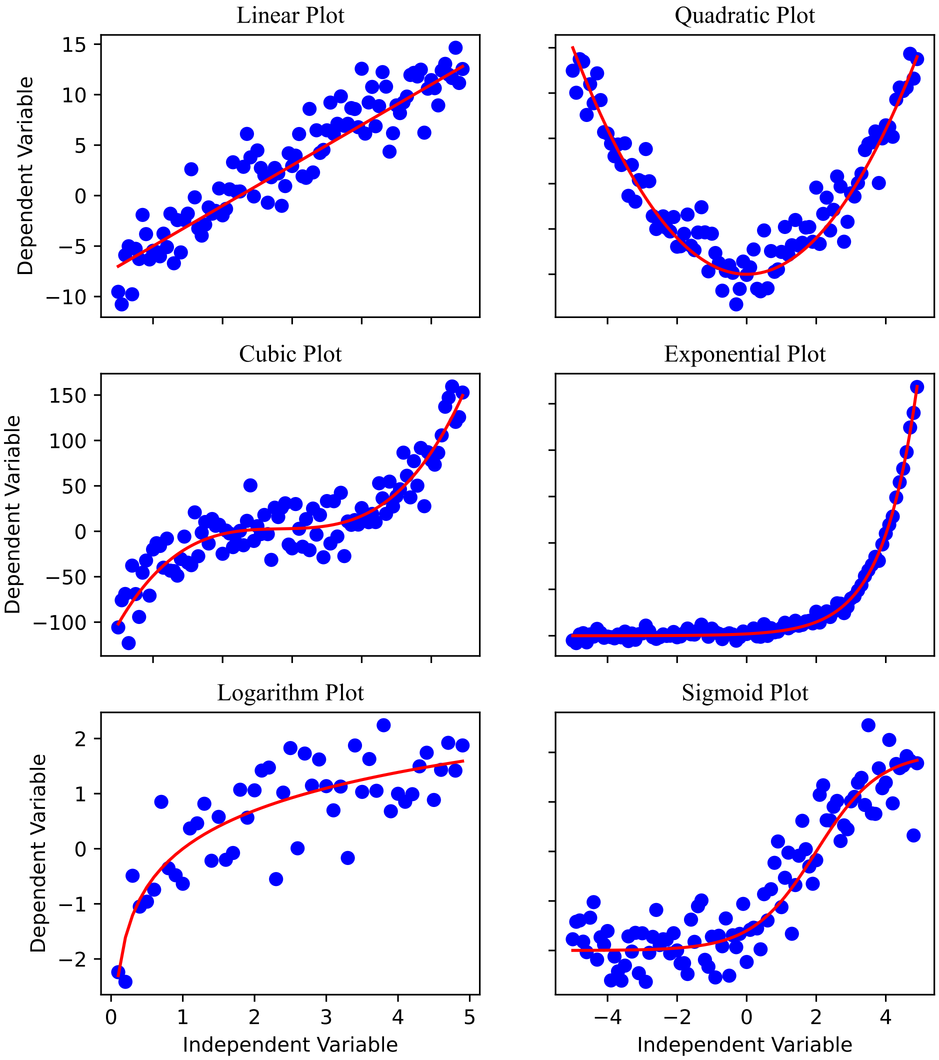
Non Linear Equations

```
#Linear, Quadratic, Cubic, Exponential, Logarithm, Sigmoidal/Logistic curve example#

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rc, font_manager
from sklearn import linear_model

ticks_font = font_manager.FontProperties(family='Times New Roman', style='normal',
    size=12, weight='normal', stretch='normal')

# Arrange x data from -5 to 5 with an interval of 0.1#

x=np.arange(-5.0,5.0,0.1)
y_noise=2*np.random.normal(size=x.size)

fig, axs = plt.subplots(3,2)
# Linear function y=ax+b #
y=2*(x)+3
ydata=y+y_noise

axs[0, 0].plot(x,ydata,'bo')
axs[0, 0].plot(x,y,color='red')
axs[0, 0].set_title('Linear Plot',fontname='Times New Roman',
          fontsize=12)

# Quadratic function y=X^2#
y_q=np.power(x,2)
ydata_q=y_q+y_noise


axs[0, 1].plot(x,ydata_q,'bo')
axs[0, 1].plot(x,y_q,color='red')
axs[0, 1].set_title('Quadratic Plot',fontname='Times New Roman',
          fontsize=12)

# Cubic function y=ax^3+bx^2+cx+d#
y_c=1*(x**3)+1*(x**2)+1*x+3
y_noise_c=20*np.random.normal(size=x.size)
ydata_c=y_c+y_noise_c

axs[1, 0].plot(x,ydata_c,'bo')
```

```
axs[1, 0].plot(x,y_c,color='red')
axs[1, 0].set_title('Cubic Plot',fontname='Times New Roman',
          fontsize=12)

# Exponential function y=a+b*c^x#
y_exp=np.exp(x)
ydata_exp=y_exp+y_noise

axs[1, 1].plot(x,ydata_exp,'bo')
axs[1, 1].plot(x,y_exp,color='red')
axs[1, 1].set_title('Exponential Plot',fontname='Times New Roman',
          fontsize=12)

# Logarithm function y=log(x)#
y_log=np.log(x)
y_noise_log=0.5*np.random.normal(size=x.size)
ydata_log=y_log+y_noise_log

axs[2, 0].plot(x,ydata_log,'bo')
axs[2, 0].plot(x,y_log,color='red')
axs[2, 0].set_title('Logarithm Plot',fontname='Times New Roman',
          fontsize=12)
# Sigmoid/Logitic function y=a+b/(1+C^(x-d))#
y_sig=1-4/(1+np.power(3,x-2))
y_noise_sig=0.5*np.random.normal(size=x.size)
ydata_sig=y_sig+y_noise_sig

axs[2, 1].plot(x,ydata_sig,'bo')
axs[2, 1].plot(x,y_sig,color='red')
axs[2, 1].set_title('Sigmoid Plot',fontname='Times New Roman',
          fontsize=12)

#Display plot#
for ax in axs.flat:
    ax.set(xlabel='Independent Variable', ylabel='Dependent Variable')
for ax in axs.flat:
    ax.label_outer()
plt.show()
```

Solution:


Linear Plot
Quadratic Plot
Cubic Plot
Exponential Plot
Logarithm Plot
Sigmoid Plot
Dependent Variable
Dependent Variable
Dependent Variable
15
10
5
0
−5
−10
150
100
50
0
−50
−100
2
1
0
−1
−2
0
1
2
3
4
5
−4
−2
0
2
4
Independent Variable
Independent Variable