Linear, Non-Linear Examples

Objective: visualize linear, quadratic, cubic, exponential, log, sine functions on a plot

In [1]:
```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
```

In [2]:
```
def log_func(x):
    if x > 0:
        return math.log(x,10)
    else:
        return np.nan
```

In [3]:
```
df = pd.DataFrame(index=range(-200,200))
```

In [4]:
```
df.shape
```

Out[4]:
```
(400, 0)
```

In [5]:
```
title = ['Linear 3*X','Quadratic X^2',
         'Cubic X^3','Absolute abs(X)',
         'sine(X)','log(X)',
         'Exponential 2^X']

df['linear']=df.index.map(lambda x: 3*x)
df['quadratic'] = df.index.map(lambda x: x**2)
df['cubic'] = df.index.map(lambda x: x**3)
df['abs'] = df.index.map(lambda x: abs(x))
df['sine'] = np.sin(np.arange(-20,20,.1))
df['log'] = df.index.map(log_func)
df['exponential'] = df.index.map(lambda x: 2.0**x)
```

In [6]:
```
df.head()
```

Out[6]:

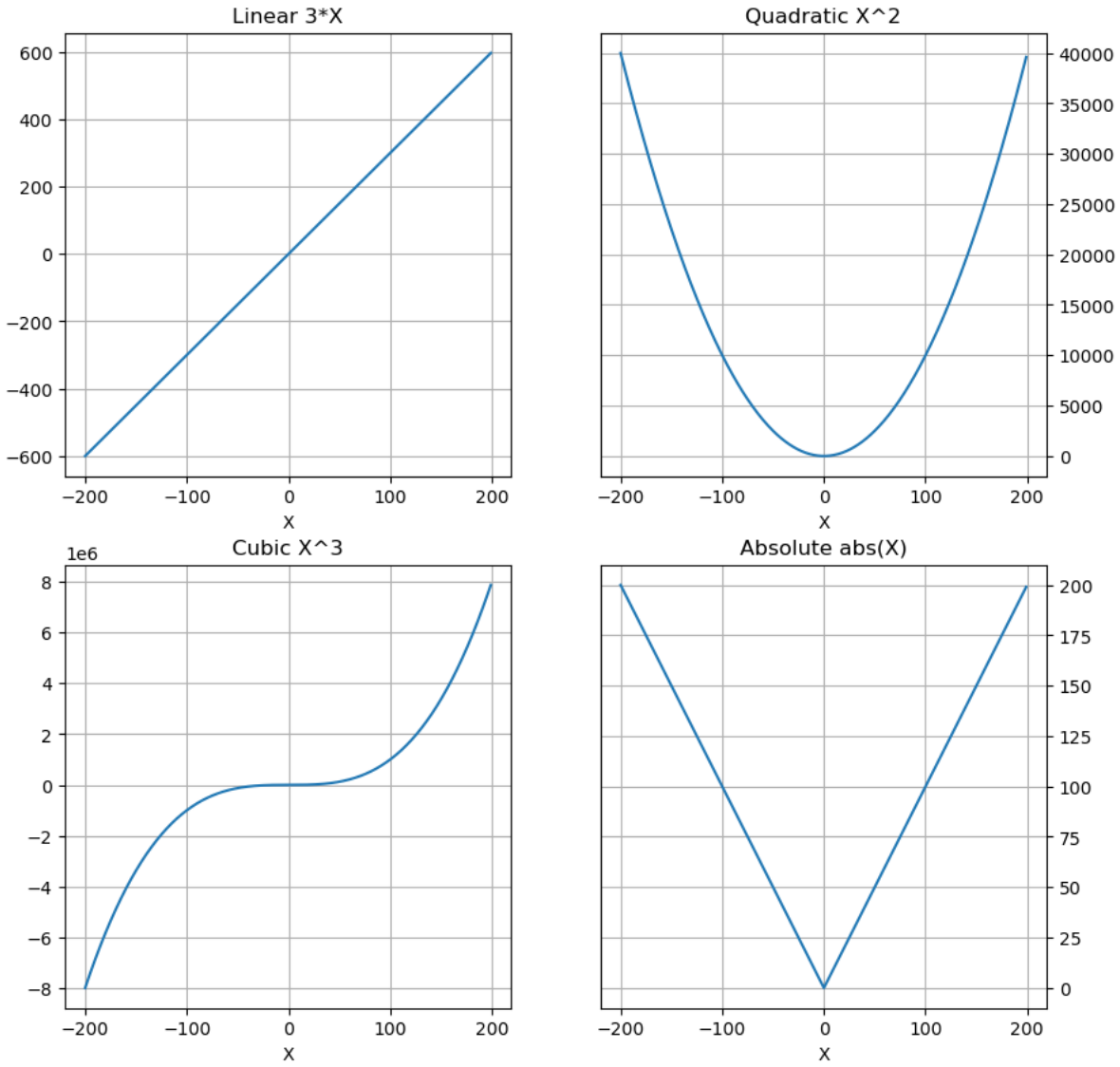
	linear	quadratic	cubic	abs	sine	log	exponential
-200	-600	40000	-8000000	200	-0.912945	NaN	6.223015e-61
-199	-597	39601	-7880599	199	-0.867644	NaN	1.244603e-60
-198	-594	39204	-7762392	198	-0.813674	NaN	2.489206e-60
-197	-591	38809	-7645373	197	-0.751573	NaN	4.978412e-60
-196	-588	38416	-7529536	196	-0.681964	NaN	9.956824e-60

In [7]:
```
df.tail()
```

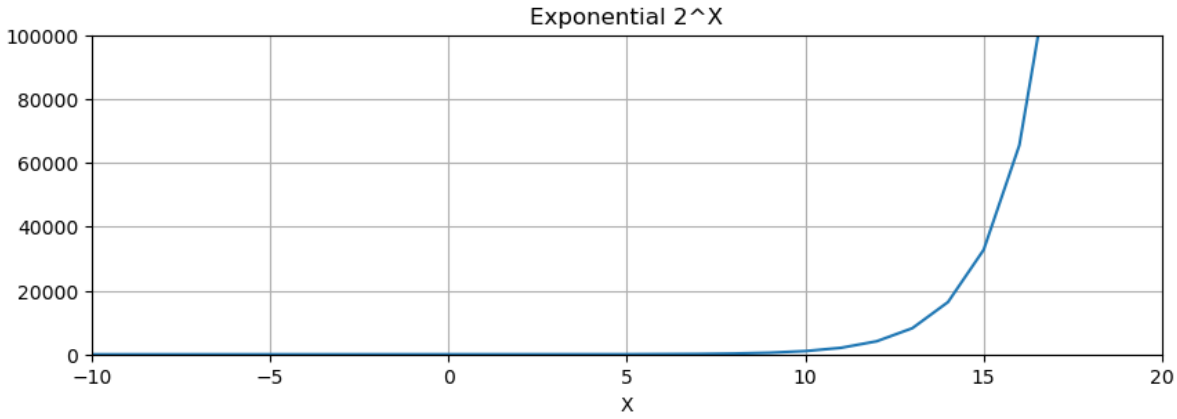
Out[7]:

	linear	quadratic	cubic	abs	sine	log	exponential
195	585	38025	7414875	195	0.605540	2.290035	5.021681e+58
196	588	38416	7529536	196	0.681964	2.292256	1.004336e+59
197	591	38809	7645373	197	0.751573	2.294466	2.008673e+59
198	594	39204	7762392	198	0.813674	2.296665	4.017345e+59
199	597	39601	7880599	199	0.867644	2.298853	8.034690e+59

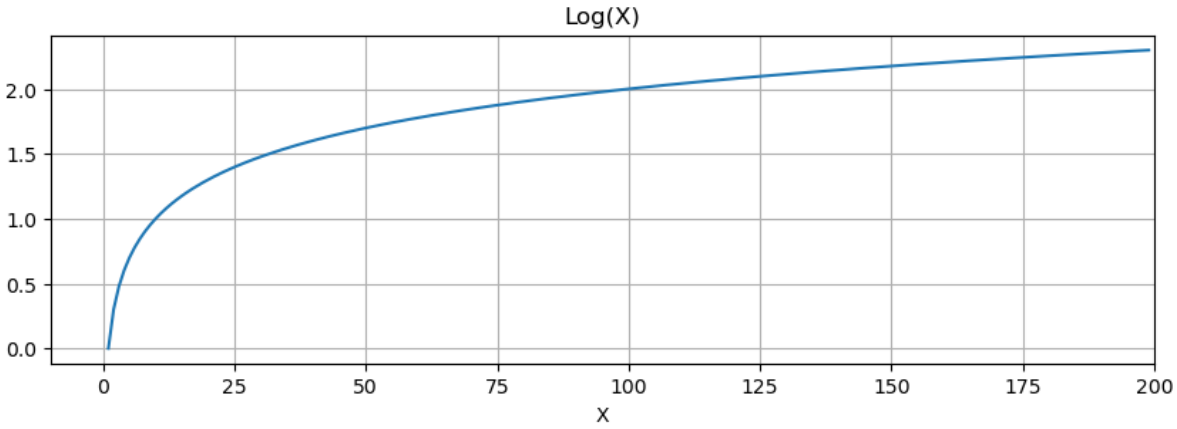
In [8]:
```
fig, axs = plt.subplots(2, 2, figsize=(10, 10), sharex=False)
axx = axs.ravel()
for i in range(df.shape[1]-3):
    axx[i].set_title(title[i])
    df[df.columns[i]].plot(ax=axx[i])
    axx[i].set_xlabel('X')
    if i % 2 == 1 :
        axx[i].yaxis.tick_right()
    axx[i].grid()
```



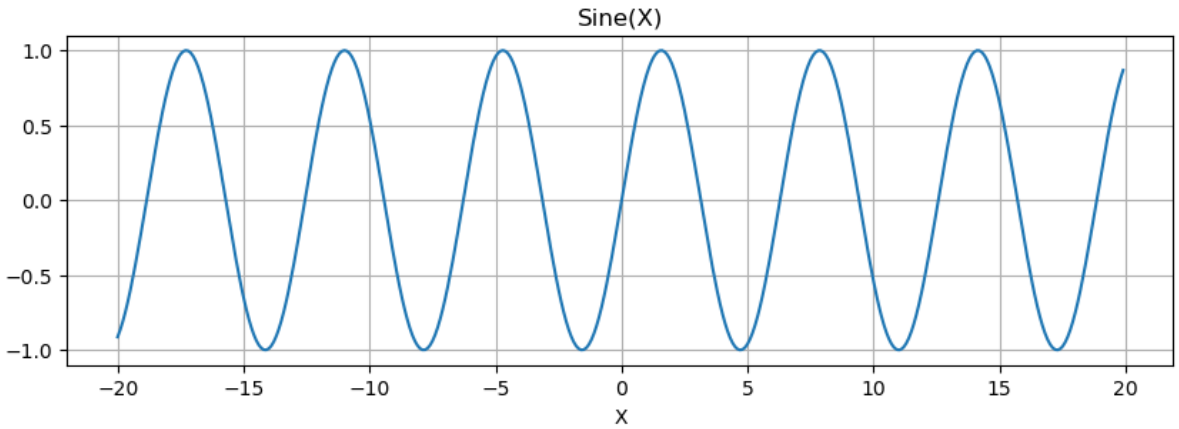
In [9]:
```
plt.figure(figsize=(10,3))
plt.plot(df['exponential'])
plt.title('Exponential 2^X')
plt.xlim(-10,20)
plt.ylim(0,100000)
plt.xlabel('X')
plt.grid()
```



In [10]:
```
plt.figure(figsize=(10,3))
plt.plot(df['log'])
plt.title('Log(X)')
plt.xlim(-10,200)
plt.xlabel('X')
plt.grid()
```



In [11]:
```
plt.figure(figsize=(10,3))
plt.plot(np.arange(-20,20,.1), df['sine'])
plt.title('Sine(X)')
plt.xlabel('X')
plt.grid()
```



In [ ]: