Integral_Line_Stock_Histocial_Data

September 29, 2021

1 Intergal using Line Equation from Stock Histocial Data

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sympy import *

import warnings
warnings.filterwarnings("ignore")

# yfinance is used to fetch data
import yfinance as yf
yf.pdr_override()
```

[2]:
```
# input
symbol = 'AMD'
start = '2017-01-01'
end = '2019-01-01'

# Read data
dataset = yf.download(symbol,start,end)['Adj Close']

# View Columns
dataset.head()
```

```
[**********100%**********]  1 of 1 completed
```

[2]:
```
Date
2017-01-03    11.43
2017-01-04    11.43
2017-01-05    11.24
2017-01-06    11.32
2017-01-09    11.49
Name: Adj Close, dtype: float64
```

[3]:
```
df = dataset.reset_index()
```

[4]:
```
df.head()
```

[4]:
```
        Date  Adj Close
0 2017-01-03      11.43
1 2017-01-04      11.43
2 2017-01-05      11.24
3 2017-01-06      11.32
4 2017-01-09      11.49
```

[5]:
```
df.tail()
```

[5]:
```
           Date  Adj Close
497 2018-12-24  16.650000
498 2018-12-26  17.900000
499 2018-12-27  17.490000
500 2018-12-28  17.820000
501 2018-12-31  18.459999
```

[6]:
```
max_p = df['Adj Close'].max()
min_p = df['Adj Close'].min()
avg_p = df['Adj Close'].mean()
```

[7]:
```
data = df.drop(['Date'], axis=1)
data
```

[7]:
```
     Adj Close
0    11.430000
1    11.430000
2    11.240000
3    11.320000
4    11.490000
..         …
497  16.650000
498  17.900000
499  17.490000
500  17.820000
501  18.459999

[502 rows x 1 columns]
```

[8]:
```
data = data.reset_index()
```

[9]:
```
data.values
```

[9]:
```
array([[  0.        ,  11.43000031],
       [  1.        ,  11.43000031],
       [  2.        ,  11.23999977],
```

```
       …,
       [499.        ,  17.48999977],
       [500.        ,  17.81999969],
       [501.        ,  18.45999908]])
```

[10]:
```
from numpy import ones,vstack
from numpy.linalg import lstsq
```

[11]:
```
points = data.values
```

[12]:
```
x_coords, y_coords = zip(*points)
A = vstack([x_coords,ones(len(x_coords))]).T
m, c = lstsq(A, y_coords)[0]
```

[13]:
```
print("Line Equation is y = {m}x + {c}".format(m=m,c=c))
```

```
Line Equation is y = 0.021718614923358828x + 9.372574584656501
```

[14]:
```
equation_of_line = print("y = {m}x + {c}".format(m=m,c=c))
```

```
y = 0.021718614923358828x + 9.372574584656501
```

[15]:
```
equation = print("{m}*x + {c}".format(m=m,c=c))
```

```
0.021718614923358828*x + 9.372574584656501
```

[16]:
```
x = Symbol('x')
```

[17]:
```
integrate(0.021718614923358828*x+9.372574584656501, x)
```

[17]: $0.0108593074616794x^2 + 9.3725745846565x$

[18]:
```
integrate(0.0108593074616794*x**2 + 9.3725745846565 * x, x)
```

[18]: $0.00361976915389313x^3 + 4.68628729232825x^2$

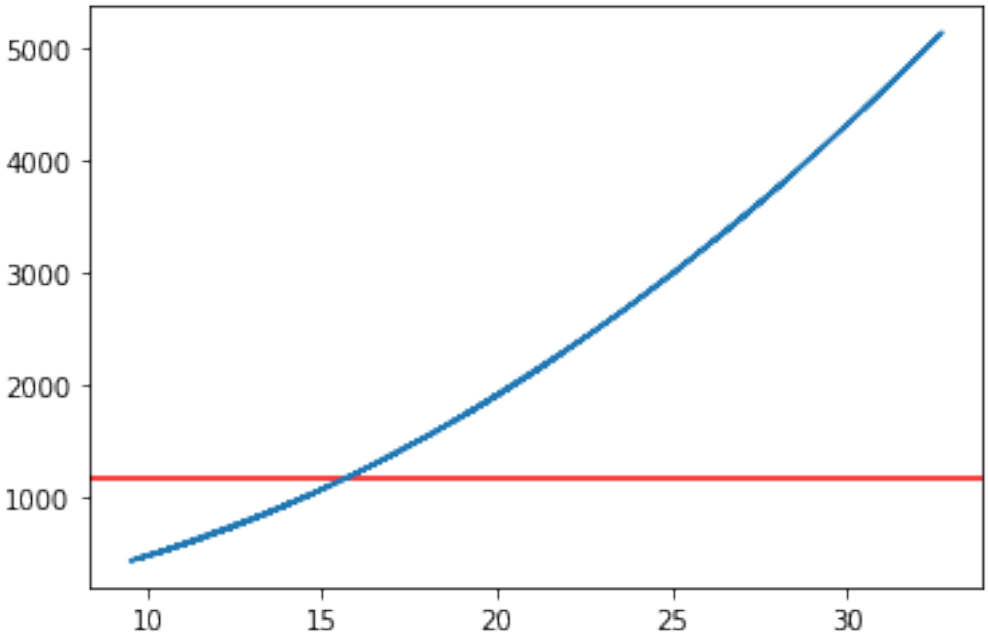
2 Univariate roots and fixed points

[19]:
```
def f(x):
    return 0.00361976915389313*x**3 + 4.68628729232825 * x**2
```

[20]:
```
x = df['Adj Close']
```

[21]:
```
plt.axhline((f(x)).mean(), c='red')
plt.plot(x, f(x))
```

[21]:
```
[<matplotlib.lines.Line2D at 0x19d34798320>]
```


5000
4000
3000
2000
1000
10
15
20
25
30