

parameter_estimation

June 9, 2023

```
[6]: # Ejemplo tomado de

#https://medium.com/@amirarsalan.rajabi/
#distribution-fitting-with-python-scipy-bb70a42c0aed

import pandas as pd
df = pd.read_csv("dow_jones_index.data")
df.head()
```

```
[6]:  quarter stock      date  open  high  low  close  volume \
0      1    AA  1/7/2011  $15.82  $16.72  $15.78  $16.42  239655616
1      1    AA  1/14/2011  $16.71  $16.71  $15.64  $15.97  242963398
2      1    AA  1/21/2011  $16.19  $16.38  $15.60  $15.79  138428495
3      1    AA  1/28/2011  $15.87  $16.63  $15.82  $16.13  151379173
4      1    AA   2/4/2011  $16.18  $17.39  $16.18  $17.14  154387761

    percent_change_price  percent_change_volume_over_last_wk \
0                3.79267                                NaN
1               -4.42849                                1.380223
2               -2.47066                               -43.024959
3                1.63831                                9.355500
4                5.93325                                1.987452

    previous_weeks_volume  next_weeks_open  next_weeks_close \
0                NaN                $16.71                $15.97
1          239655616.0                $16.19                $15.79
2          242963398.0                $15.87                $16.13
3          138428495.0                $16.18                $17.14
4          151379173.0                $17.33                $17.37

    percent_change_next_weeks_price  days_to_next_dividend \
0                -4.428490                                26
1                -2.470660                                19
2                 1.638310                                12
3                 5.933250                                 5
4                 0.230814                                97
```

```

percent_return_next_dividend
0          0.182704
1          0.187852
2          0.189994
3          0.185989
4          0.175029

```

```
[8]: # suprimimos los dignos de dolar porque sólo queremos valores numéricos
```

```

def omit_s(x):
    return x[1:]
df['high'] = df['high'].apply(omit_s)
df['low'] = df['low'].apply(omit_s)
df['open'] = df['open'].apply(omit_s)
df['close'] = df['close'].apply(omit_s)
df.head()

```

```

[8]:  quarter stock      date  open  high  low  close  volume \
0      1    AA  1/7/2011  15.82  16.72  15.78  16.42  239655616
1      1    AA  1/14/2011  16.71  16.71  15.64  15.97  242963398
2      1    AA  1/21/2011  16.19  16.38  15.60  15.79  138428495
3      1    AA  1/28/2011  15.87  16.63  15.82  16.13  151379173
4      1    AA  2/4/2011  16.18  17.39  16.18  17.14  154387761

```

```

percent_change_price  percent_change_volume_over_last_wk \
0          3.79267          NaN
1         -4.42849          1.380223
2         -2.47066         -43.024959
3          1.63831          9.355500
4          5.93325          1.987452

```

```

previous_weeks_volume  next_weeks_open  next_weeks_close \
0          NaN          $16.71          $15.97
1      239655616.0          $16.19          $15.79
2      242963398.0          $15.87          $16.13
3      138428495.0          $16.18          $17.14
4      151379173.0          $17.33          $17.37

```

```

percent_change_next_weeks_price  days_to_next_dividend \
0          -4.428490          26
1          -2.470660          19
2           1.638310          12
3           5.933250           5
4           0.230814          97

```

```

percent_return_next_dividend
0          0.182704

```

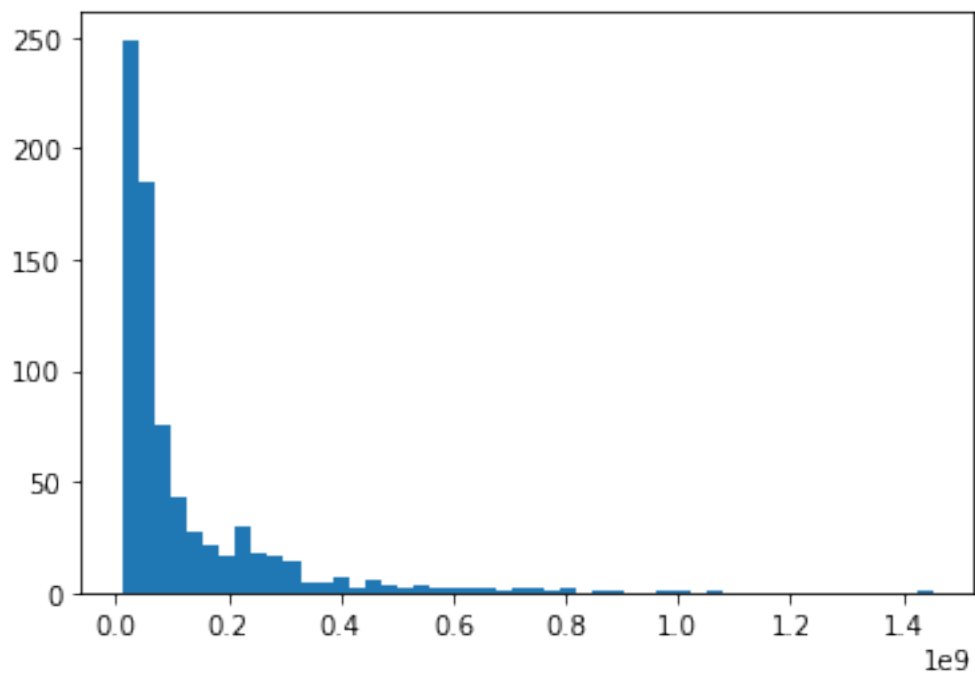
1	0.187852
2	0.189994
3	0.185989
4	0.175029

```
[9]: df.shape
```

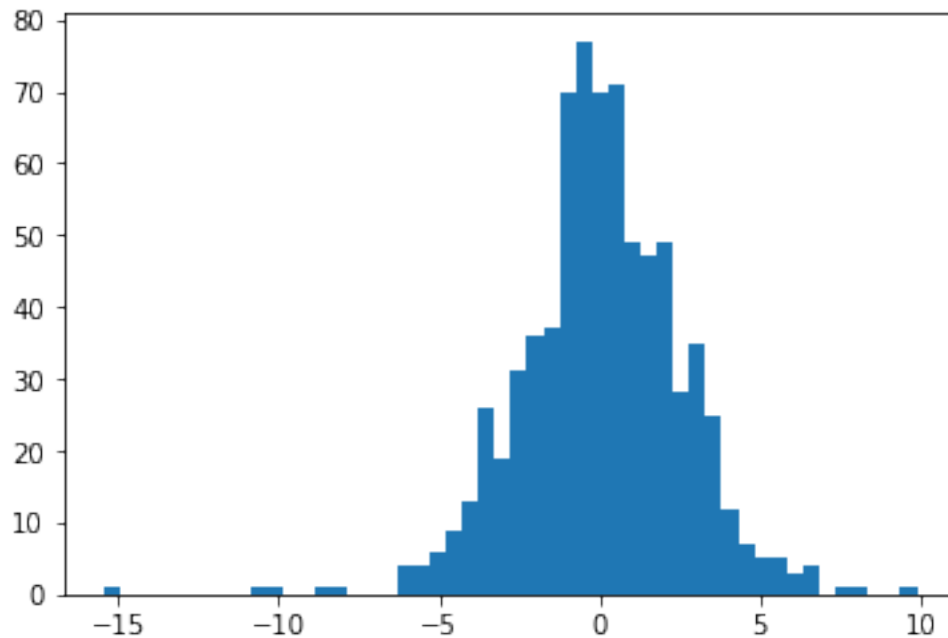
```
[9]: (750, 16)
```

```
[16]: # Hacemos un histograma de los volúmenes negociados...
      # No parecen tener una distribución normal
```

```
import matplotlib.pyplot as plt
plt.hist(df['volume'], bins=50)
plt.show()
```



```
[15]: # Ahora con el porcentaje de cambio en el precio
      # ¡Se parece a una normal !
plt.hist(df['percent_change_price'], bins=50)
plt.show()
```



```
[24]: # Estimamos los parámetros óptimos usando máxima verosimilitud
from scipy import stats
distribution=stats.norm
parametros=distribution.fit(df['percent_change_price'])
parametros
```

```
[24]: (0.050262410000000003, 2.516130107690429)
```

```
[41]: # Comparamos gráficamente la distribución de los datos con la obtenida

distribution_with_parameters=stats.norm(loc=parametros[0],scale=parametros[1])
import numpy as np
x=np.linspace(start=-15,stop=10,num=100)
y=distribution_with_parameters.pdf(x)
plt.plot(x,y,"red")
plt.hist(df['percent_change_price'], bins=50,density=True)
plt.show()
```

