/opt/pyenv-3.7.5/bin/jupyter notebook

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

pandas' describe() function

In [2]:
```
x = 1.0 + 2.0 * np.random.randn(1000)**3
```

In [3]:
```
dataset = pd.DataFrame(x)
```

In [4]:
```
dataset.head()
```

Out[4]:

	0
0	1.960967
1	-1.442851
2	8.998921
3	-4.603361
4	10.689080

In [5]:
```
dataset.describe()
```

Out[5]:

	0
count	1000.000000
mean	1.339696
std	8.335210
min	-74.196589
25%	0.504930
50%	1.000000
75%	1.608701
max	79.355883

$$\text{count} \leftrightarrow N$$
$$\text{mean} \leftrightarrow \bar{X},\ \text{std} \leftrightarrow s$$
$$25\% \leftrightarrow Q_1,\ 50\% \leftrightarrow M \equiv Q_2,\ 75\% \leftrightarrow Q_3$$

Μέτρα Ασυμμετρίας

Μέτρο ασυμμετρίας Pearson

$$\tilde{Sk}_p = \frac{3(\bar{X} - M)}{s}$$

In [6]:
```
pearson = 3*(dataset.mean()-dataset.median())/dataset.std()
```

In [7]:
```
pearson
```

Out[7]:
```
0    0.122263
dtype: float64
```

Μέτρο ασυμμετρίας Bowley

$$Sk_b = \frac{(Q_3 - M) - (M - Q_1)}{Q_3 - Q_1}$$

In [8]:
```
Q_1 = dataset.quantile(0.25)
M = dataset.median()
Q_3 = dataset.quantile(0.75)
bowley = ((Q_3 - M) - (M - Q_1))/(Q_3-Q_1)
```

In [9]:
```
bowley
```

Out[9]:
```
0    0.102948
dtype: float64
```

Καμπύλη Lorenz - Συντελεστής Gini

In [10]:
```
wage = np.array([500, 1000, 1000, 1000, 2000, 2000])

wage = np.insert(wage, 0, 0.0)
```

In [11]:
```
df = pd.DataFrame(wage, columns=['x'])
```

In [12]:
```
df.describe()
```

Out[12]:

	x
count	7.000000
mean	1071.428571
std	731.925055
min	0.000000
25%	750.000000
50%	1000.000000
75%	1500.000000
max	2000.000000

$$\Phi_n = \frac{\sum_{j=1}^{n} x_j}{\sum_{j=1}^{N} x_j}$$

In [13]:
```
df['Phi'] = df['x'].cumsum()/df['x'].sum()
```

In [14]:
```
df['x']
```

Out[14]:
```
0       0
1     500
2    1000
3    1000
4    1000
5    2000
6    2000
Name: x, dtype: int64
```

In [15]:
```
df['x'].cumsum()
```

Out[15]:
```
0       0
1     500
2    1500
3    2500
4    3500
5    5500
6    7500
Name: x, dtype: int64
```

In [16]:
```
df['x'].sum()
```

Out[16]:
```
7500
```

In [17]:
```
df['Phi']
```

Out[17]:
```
0    0.000000
1    0.066667
2    0.200000
3    0.333333
4    0.466667
5    0.733333
6    1.000000
Name: Phi, dtype: float64
```

$$RF_n = \frac{n}{N}$$

In [18]:
```
df['RF'] = np.array(range(len(wage)))/(len(wage)-1)
```

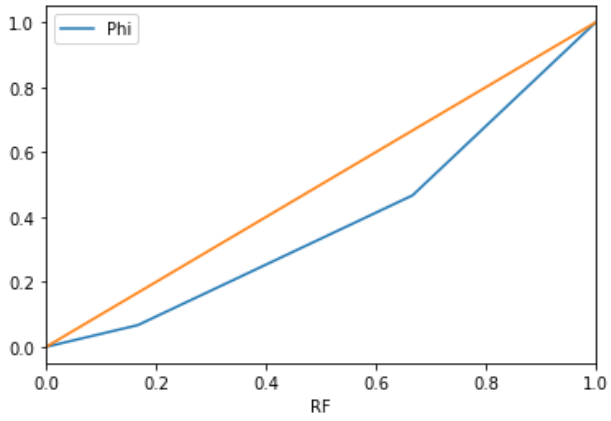
In [19]:
```
df['RF']
```

Out[19]:
```
0    0.000000
1    0.166667
2    0.333333
3    0.500000
4    0.666667
5    0.833333
6    1.000000
Name: RF, dtype: float64
```

$$\{(RF_n, \Phi_n)\}_{n=1}^{N}$$

In [20]:
```
df.plot(x='RF',y='Phi')
plt.plot((0,1),(0,1))
```

Out[20]:
```
[<matplotlib.lines.Line2D at 0x7f4ad3b61a90>]
```



In [21]:
```
df
```

Out[21]:

	x	Phi	RF
0	0	0.000000	0.000000
1	500	0.066667	0.166667
2	1000	0.200000	0.333333
3	1000	0.333333	0.500000
4	1000	0.466667	0.666667
5	2000	0.733333	0.833333
6	2000	1.000000	1.000000

In [22]:
```
df['Phi'][1:].values
```

Out[22]:
```
array([0.06666667, 0.2       , 0.33333333, 0.46666667, 0.73333333,
       1.        ])
```

In [23]:
```
df['Phi'][:-1].values
```

Out[23]:
```
array([0.        , 0.06666667, 0.2       , 0.33333333, 0.46666667,
       0.73333333])
```

$$\Sigma\Phi_n = \Phi_n + \Phi_{n-1}$$

In [24]:
```
SPhi = df['Phi'][1:].values + df['Phi'][:-1].values
```

In [25]:
```
SPhi = np.insert(SPhi, 0, 0.0)
```

In [26]:
```
df['SPhi'] = SPhi
```

In [27]:
```
df
```

Out[27]:

	x	Phi	RF	SPhi
0	0	0.000000	0.000000	0.000000
1	500	0.066667	0.166667	0.066667
2	1000	0.200000	0.333333	0.266667
3	1000	0.333333	0.500000	0.533333
4	1000	0.466667	0.666667	0.800000
5	2000	0.733333	0.833333	1.200000
6	2000	1.000000	1.000000	1.733333

$$\Delta RF_n = RF_n - RF_{n-1}$$

In [28]:
```
DRF = df['RF'][1:].values - df['RF'][0:-1].values
```

In [29]:
```
DRF = np.insert(DRF, 0, 0.0)
```

In [30]:
```
df['DRF'] = DRF
```

In [31]:
```
df['SPhi_DRF'] = df['SPhi'] * df['DRF']
```

In [32]:
```
df
```

Out[32]:

	x	Phi	RF	SPhi	DRF	SPhi_DRF
0	0	0.000000	0.000000	0.000000	0.000000	0.000000
1	500	0.066667	0.166667	0.066667	0.166667	0.011111
2	1000	0.200000	0.333333	0.266667	0.166667	0.044444
3	1000	0.333333	0.500000	0.533333	0.166667	0.088889
4	1000	0.466667	0.666667	0.800000	0.166667	0.133333
5	2000	0.733333	0.833333	1.200000	0.166667	0.200000
6	2000	1.000000	1.000000	1.733333	0.166667	0.288889

In [33]:
```
Gini = 1 - df['SPhi_DRF'].sum()
```

In [34]:
```
Gini
```

Out[34]:
```
0.23333333333333328
```

In [ ]:
```

```