NIM/Nama : 13517050/Christopher Billy Setiawan, 13517137/Vincent Budianto

Nama file : T1-IF2122-13517050-13517137.ipynb

Topik : Tugas Besar 01 IF2122 - Probabilitas dan Statistika

Tanggal : 09 April 2019

Deskripsi : Pemrosesan data statistika

In [1]:

```
from scipy import integrate
import matplotlib as plt
import math
import numpy as np
import pandas as pd
```

A. Dataset1 (fifa.csv)

In [2]:

```
data1 = pd.read_csv('fifa.csv')
```

1. Data Visualization

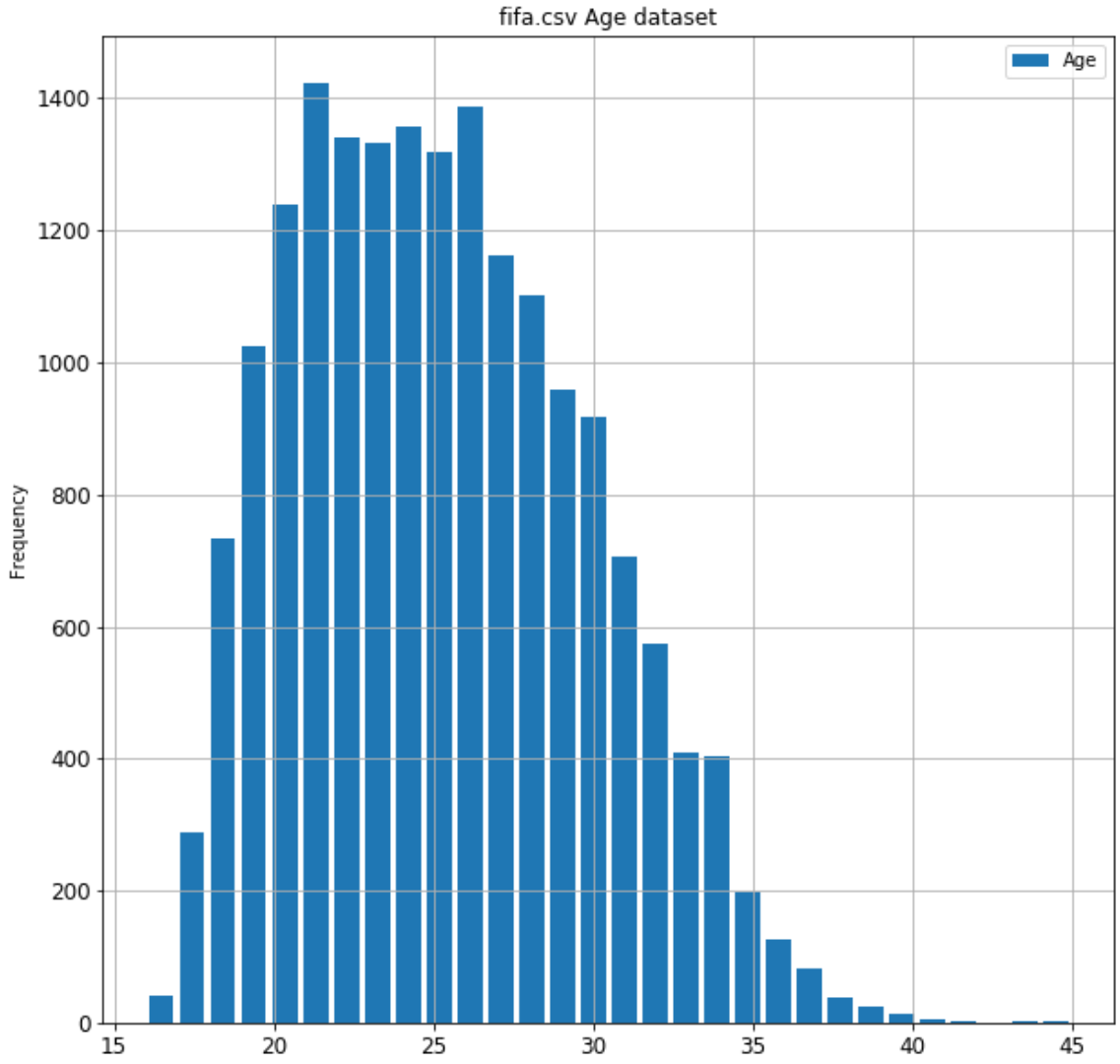
a. Histogram

In [4]:

```
data1['Age'].plot(kind = 'hist', bins = 30, rwidth = 0.8, fontsize = 'large', figsize =
(10, 10), title = 'fifa.csv Age dataset', grid = True, legend = True)
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25b62310>
```



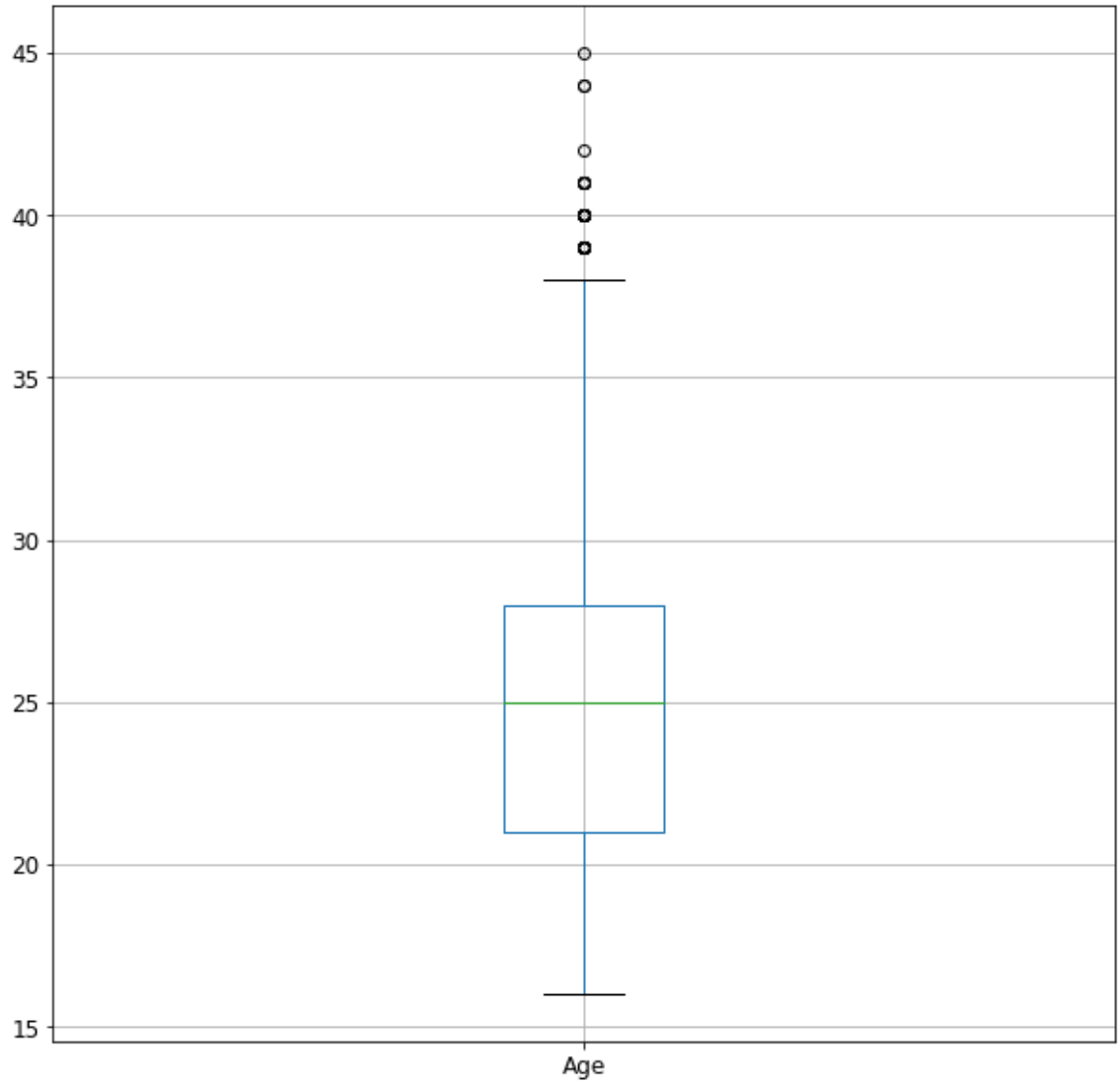
b. Box Plot

In [5]:

```
data1.boxplot(fontsize = 'large', figsize = (10, 10), grid = True)
```

Out[5]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25b9a150>
```



2. Statistical Descriptions

a. Minimum Value

In [6]:

```
data1.min(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[6]:

```
' Age  16'
```

b. Maximum Value

In [7]:

```
data1.max(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[7]:

```
' Age  45'
```

c. Mean

In [8]:

```
data1.mean(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[8]:

```
' Age  25.122'
```

d. Mode

In [9]:

```
data1.mode(numeric_only = True).round(3).to_string(index = None)
```

Out[9]:

```
' Age\n  21'
```

e. Median

In [10]:

```
data1.median(numeric_only = True).round(3).reset_index().to_string(header = None, index
= None)
```

Out[10]:

```
' Age  25.0'
```

f. Variance

In [11]:

```
data1.var(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[11]:

```
' Age  21.808'
```

g. Standard Deviation

In [12]:

```
data1.std(numeric_only = True).round(3).reset_index().to_string(header = None, index = None)
```

Out[12]:

```
' Age  4.67'
```

h. Skewness

In [13]:

```
data1.skew(numeric_only = True).round(3).reset_index().to_string(header = None, index = None)
```

Out[13]:

```
' Age  0.392'
```

i. Kurtosis

In [14]:

```
data1.kurt(numeric_only = True).round(3).reset_index().to_string(header = None, index = None)
```

Out[14]:

```
' Age -0.46'
```

3. Distribution Function

Dari data kasus yang diajukan, dapat kami simpulkan bahwa persebaran data bersifat diskrit dan penggunaan fungsi distribusi yang cocok hanya fungsi distribusi binomial atau fungsi distribusi hipergeometri, karena selain dari kedua fungsi itu tidak ada fungsi distribusi yang cocok dengan kasus ini. Lalu pertanyaan yang diberikan hanya memiliki 2 kemungkinan yaitu antara salah atau benar dan setiap 1000 pemain yang diambil oleh Tsubasa tidak ada yang sama. Dari sana kami dapat menyimpulkan dalam kasus ini yang paling tepat adalah menggunakan fungsi distribusi hipergeometri. Karena jika menggunakan binomial, sample yang diambil akan dikembalikan sehingga ada kemungkinan sukses setelah mengambil lebih dari banyak jumlah data total.

4. Questions

a. Jika terdapat 1000 pemain bola baru yang ditambahkan oleh Tsubasa, tentukan ekspektasi umur pemain bola yang:

i. Berumur kurang dari 22 tahun

In [15]:

```
round((len(data1.loc[data1['Age'] < 22]) / len(data1) * 1000))
```

Out[15]:

```
261
```

ii. Berumur lebih dari 40 tahun

In [16]:

```
round((len(data1.loc[data1['Age'] > 40]) / len(data1) * 1000))
```

Out[16]:

```
0
```

B. Dataset3 (black_friday.csv)

In [17]:

```
data3 = pd.read_csv('black_friday.csv', header = None, names = ['total'])
```

1. Data Visualization

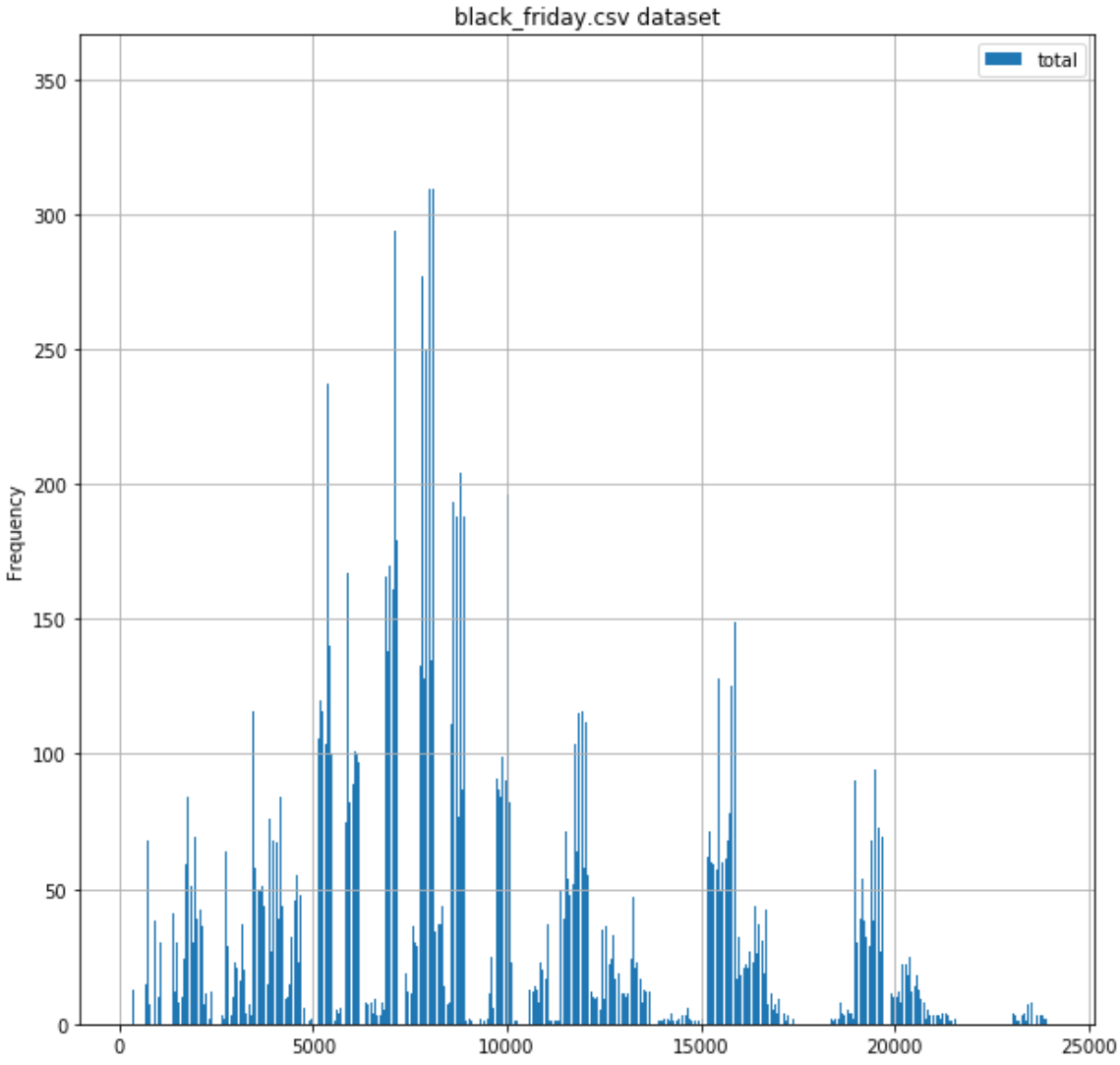
a. Histogram

In [18]:

```
data3['total'].plot(kind = 'hist', bins = 17960, rwidth = 0.8, figsize = (10, 10), titl
e = 'black_friday.csv dataset', grid = True, legend = True)
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x25fd5170>
```



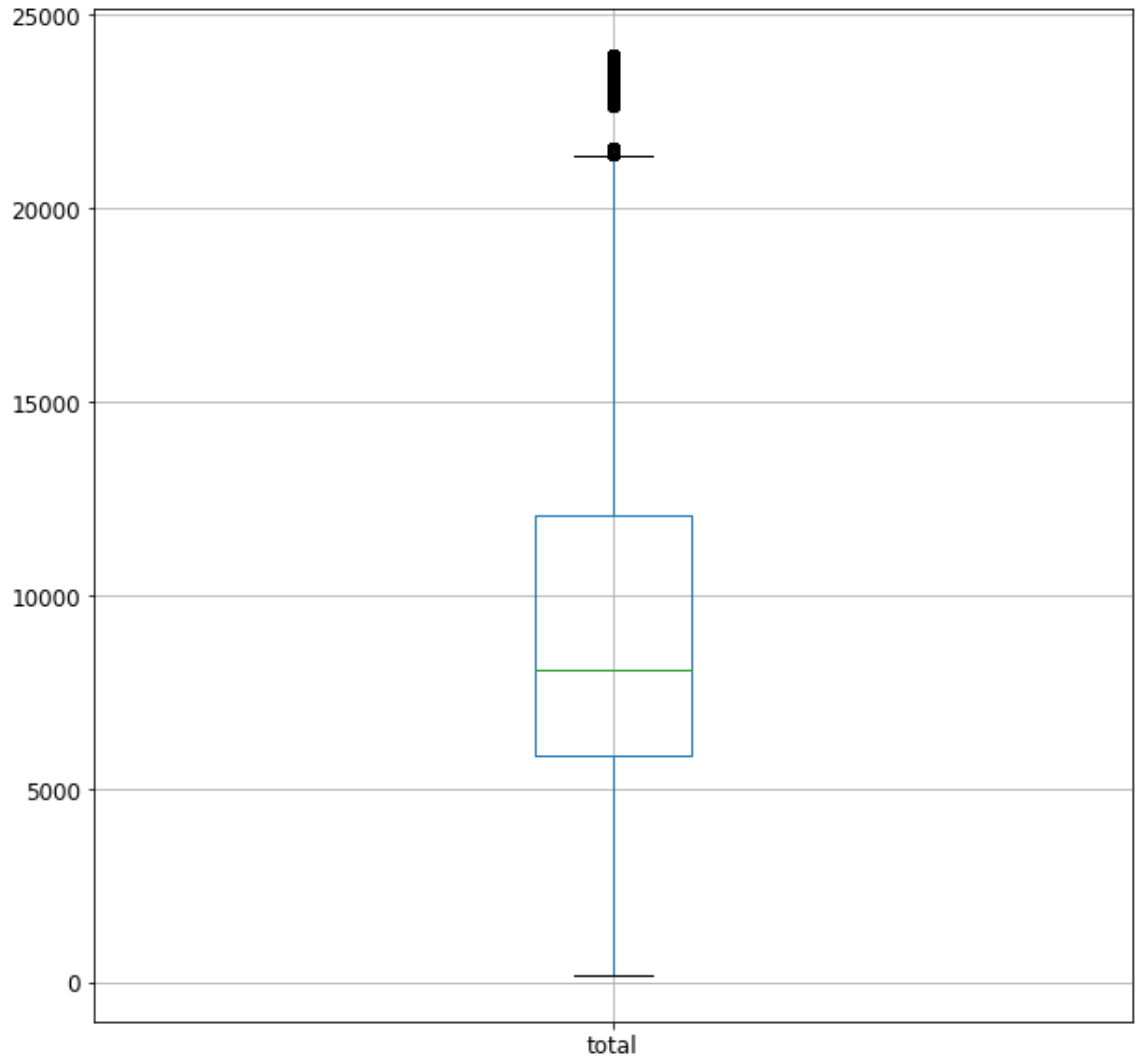
b. Box Plot

In [19]:

```
data3.boxplot(fontsize = 'large', figsize = (10, 10), grid = True)
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2dfbb6f0>
```



2. Statistical Descriptions

a. Minimum Value

In [20]:

```
data3.min(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[20]:

```
' total  185'
```

b. Maximum Value

In [21]:

```
data3.max(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[21]:

```
' total  23961'
```

c. Mean

In [22]:

```
data3.mean(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[22]:

```
' total  9333.86'
```

d. Mode

In [23]:

```
data3.mode(numeric_only = True).round(3).to_string(index = None)
```

Out[23]:

```
' total\n  6855'
```

e. Median

In [24]:

```
data3.median(numeric_only = True).round(3).reset_index().to_string(header = None, index
= None)
```

Out[24]:

```
' total  8062.0'
```

f. Variance

In [25]:

```
data3.var(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[25]:

```
' total  2.481058e+07'
```

g. Standard Deviation

In [26]:

```
data3.std(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[26]:

```
' total  4981.022'
```

h. Skewness

In [27]:

```
data3.skew(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[27]:

```
' total  0.624'
```

i. Kurtosis

In [28]:

```
data3.kurt(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[28]:

```
' total -0.343'
```

3. Distribution Function

Dari data kasus yang diajukan, dapat kami simpulkan bahwa persebaran data bersifat diskrit dan penggunaan fungsi distribusi yang cocok hanya fungsi distribusi binomial atau fungsi distribusi hipergeometri, karena selain dari kedua fungsi itu tidak ada fungsi distribusi yang cocok dengan kasus ini. Lalu pertanyaan yang diberikan hanya memiliki 2 kemungkinan yaitu antara salah atau benar. Setelah menelaah lebih lanjut, kami memilih fungsi distribusi hipergeometri. Karena jika ketika menggunakan binomial, hal ini tidak bisa dilakukan mengingat banyak data yang sama di black friday sehingga ada kemungkinan mengambil pembeli yang sama jika menggunakan binomial. Oleh karena itu kami memilih menggunakan fungsi distribusi hipergeometri.

4. Questions

a. Jika terdapat 250 orang pembeli baru yang mengikuti Black Friday, tentukan ekspektasi jumlah orang yang:

i. Miskin (total pembelian kurang dari 1000 dolar)

In [29]:

```
round(len(data3.loc[data3['total'] < 1000]) / len (data3) * 250)
```

Out[29]:

3

ii. Kaya (total pembelian lebih dari 10000 dolar)

In [30]:

```
round(len(data3.loc[data3['total'] > 10000]) / len (data3) * 250)
```

Out[30]:

87

iii. Crazy Rich (total pembelian lebih dari 20000 dolar)

In [31]:

```
round(len(data3.loc[data3['total'] > 20000]) / len (data3) * 250)
```

Out[31]:

6

b. Jika terdapat 1000 orang pembeli baru yang mengikuti Black Friday, tentukan ekspektasi jumlah orang yang sebenarnya pengeluarannya sama, seperti membeli:

i. Galaxy Fold (total pembelian di antara 1980–2000 dolar inklusif)

In [32]:

```
round(len(data3.loc[(data3['total'] >= 1980) & (data3['total'] <= 2000)]) / len (data3)
* 1000)
```

Out[32]:

1

ii. MacBook Pro 2018 Touch Bar 256GB + iPhone XR + AirPods 2 (total pembelian di antara 2707–2897 dolar inklusif)

In [33]:

```
round(len(data3.loc[(data3['total'] >= 2707) & (data3['total'] <= 2897)]) / len (data3)
* 1000)
```

Out[33]:

7

C. Dataset4 (crypto.csv)

In [34]:

```
data4 = pd.read_csv('crypto.csv', header = None, names = ['cryptocurrency'])
```

1. Data Visualization

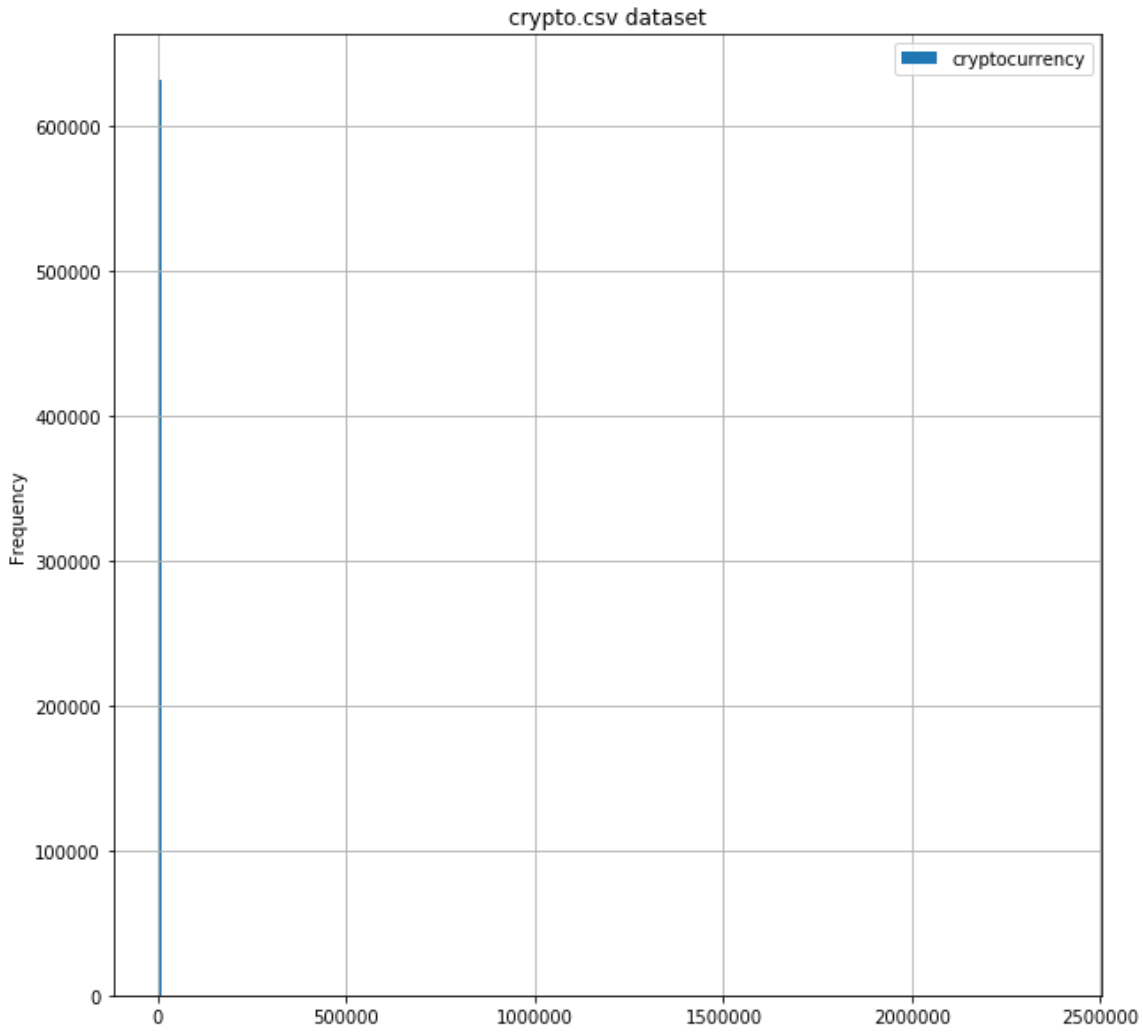
a. Histogram

In [35]:

```
data4['cryptocurrency'].plot(kind = 'hist', bins = 250, rwidth = 0.8, figsize = (10, 10
), title = 'crypto.csv dataset', grid = True, legend = True)
```

Out[35]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e11c790>
```



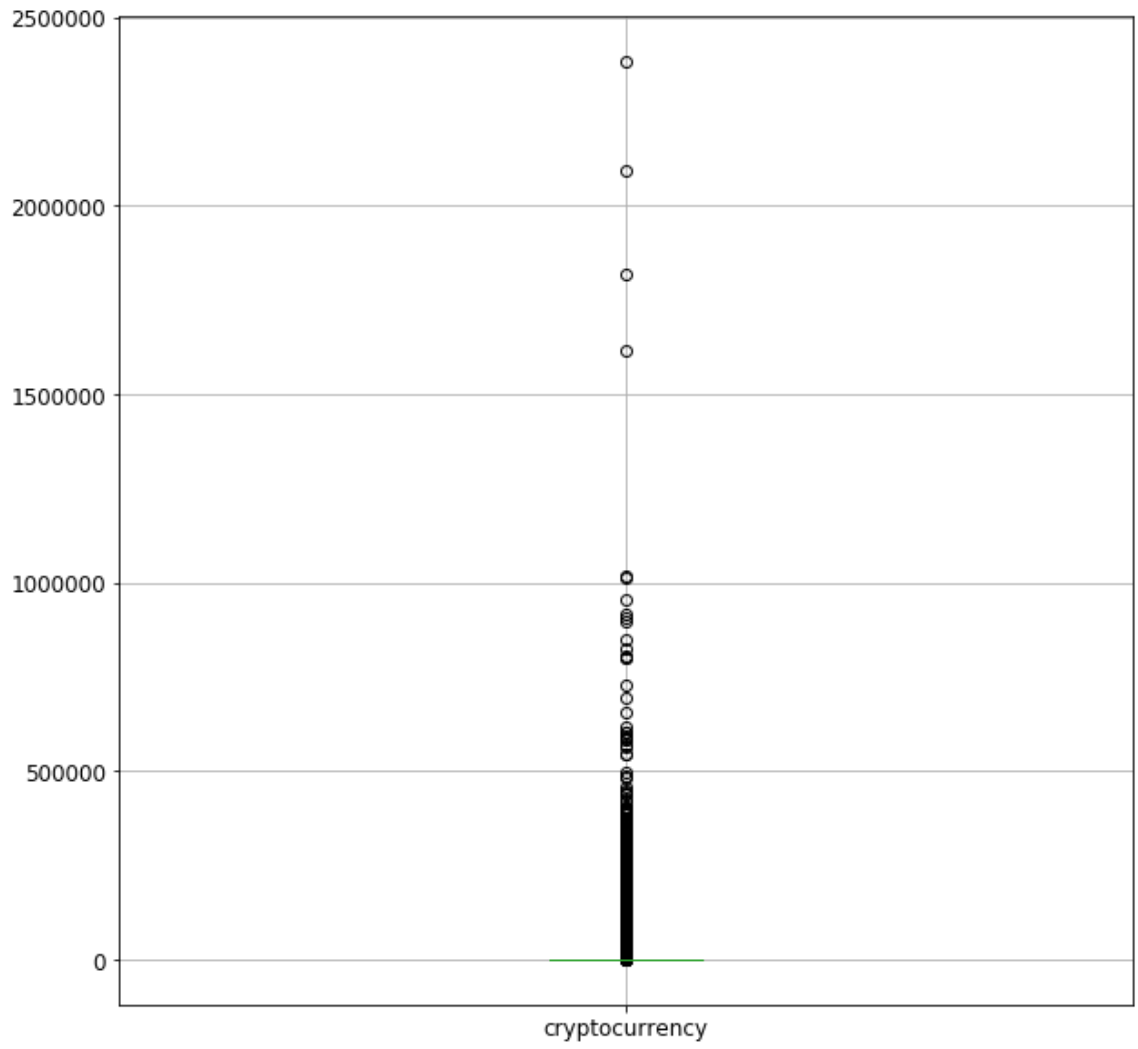
b. Box Plot

In [36]:

```
data4.boxplot(fontsize = 'large', figsize = (10, 10), grid = True)
```

Out[36]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c4bf0b0>
```



2. Statistical Descriptions

a. Minimum Value

In [37]:

```
data4.min(numeric_only = True).reset_index().to_string(header = None, index = None)
```

Out[37]:

```
' cryptocurrency  2.925000e-09'
```

b. Maximum Value

In [38]:

```
data4.max(numeric_only = True).reset_index().to_string(header = None, index = None)
```

Out[38]:

```
' cryptocurrency  2383502.5'
```

c. Mean

In [39]:

```
data4.mean(numeric_only = True).reset_index().to_string(header = None, index = None)
```

Out[39]:

```
' cryptocurrency  203.018146'
```

d. Mode

In [40]:

```
data4.mode(numeric_only = True).to_string(index = None)
```

Out[40]:

```
' cryptocurrency\n       0.000002'
```

e. Median

In [41]:

```
data4.median(numeric_only = True).reset_index().to_string(header = None, index = None)
```

Out[41]:

```
' cryptocurrency  0.009734'
```

f. Variance

In [42]:

```
data4.var(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[42]:

```
' cryptocurrency  7.532042e+07'
```

g. Standard Deviation

In [43]:

```
data4.std(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[43]:

```
' cryptocurrency  8678.734'
```

h. Skewness

In [44]:

```
data4.skew(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[44]:

```
' cryptocurrency  118.24'
```

i. Kurtosis

In [45]:

```
data4.kurt(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[45]:

```
' cryptocurrency  22297.471'
```

3. Distribution Function

Dari data kasus yang diajukan, dapat kami simpulkan bahwa persebaran data bersifat kontigu dan penggunaan fungsi distribusi yang cocok adalah fungsi distribusi gamma karena jika kita bandingkan dengan fungsi distribusi yang lain, fungsi-fungsi tersebut tidak cocok untuk diimplementasikan dalam kasus ini karena kasus ini tidak memenuhi syarat dari fungi-fungsi tersebut. Adapun ketika kami membandingkan grafik yang dihasilkan, kami menyimpulkan bahwa grafik yang dihasilkan mirip dengan grafik fungsi distribusi gamma dengan alfa = 0.000547 dan beta = 371003.365124.

4. Questions

In [46]:

```
beta = float(data4.var(numeric_only = True) / data4.mean(numeric_only = True))
alfa = float(data4.mean(numeric_only = True) / beta)

def g(x):
    h = lambda y: y**(x - 1) * np.exp(-y)
    res = integrate.quad(h, 0, np.inf)
    return res[0]

f = lambda x: (x**(alfa - 1) * np.exp(-(x / beta)) / ((beta**alfa) * g(alfa)))
```

a. Apabila hari ini terdapat 1000 data4 harga cryptocurrency baru, tentukan ekspektasi jumlah cryptocurrency yang nilainya

i. kurang dari 0.177013

In [48]:

```
a1 = integrate.quad(f, 0, 0.177013)
print(round(a1[0] * 1000))
```

```
992
```

ii. lebih dari 177.013

In [49]:

```
a2 = integrate.quad(f, 177.013, np.inf)
print(round(a2[0] * 1000))
```

```
4
```

b. Jika suatu hari terdapat sebuah cryptocurrency baru, tentukan peluang cryptocurrency tersebut bernilai

i. lebih dari 0.013

In [50]:

```
b1 = integrate.quad(f, 0.013, np.inf)
print(b1[0])
```

```
0.009037196724844068
```

ii. kurang dari 17.7

In [51]:

```
b2 = integrate.quad(f, 0, 17.7)
print(b2[0])
```

```
0.9948837253205033
```

D. Dataset5 (athletes.csv)

In [52]:

```
data5 = pd.read_csv('athletes.csv')
```

1. Data Visualization

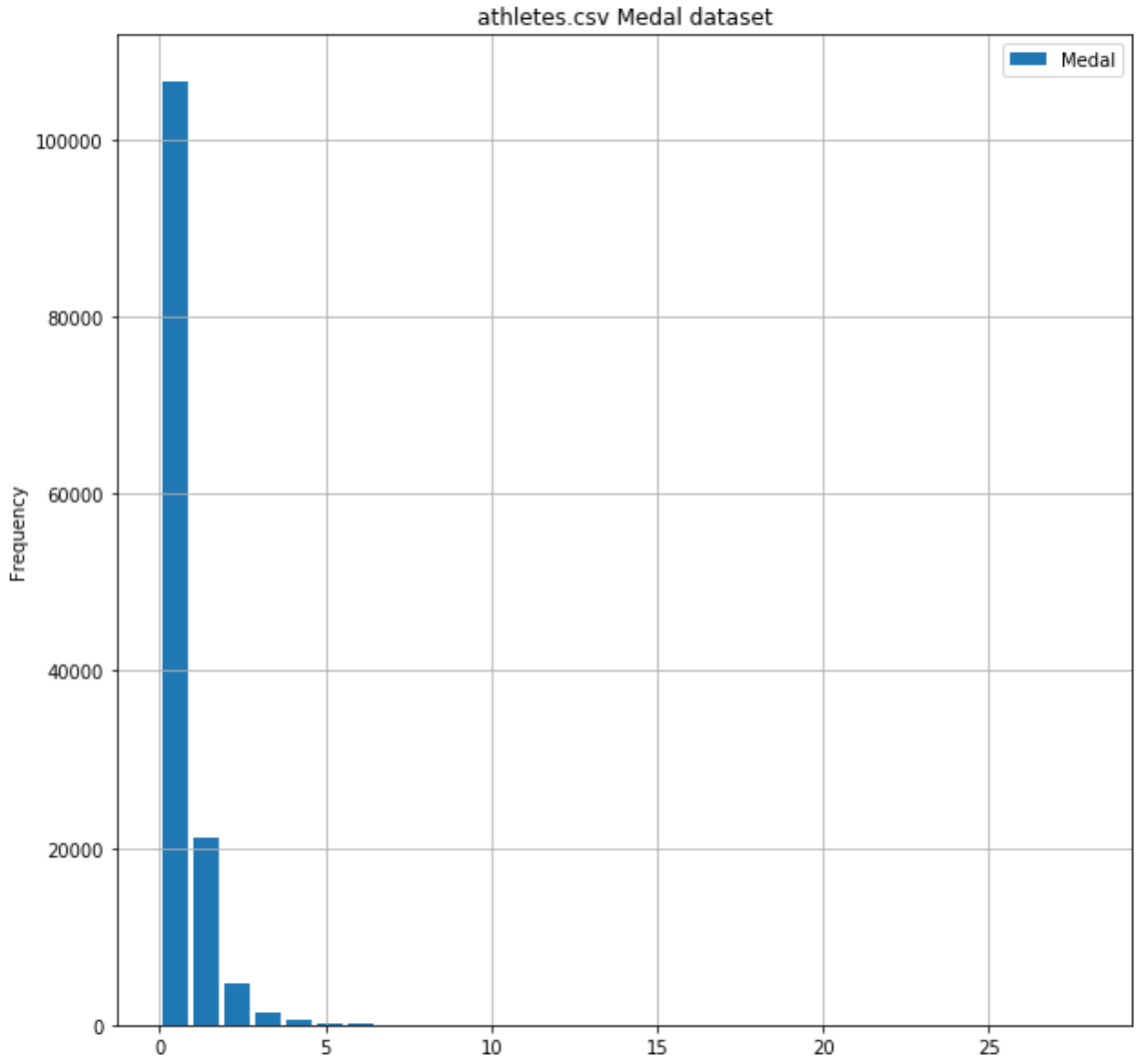
a. Histogram

In [53]:

```
data5['Medal'].plot(kind = 'hist', bins = 30, rwidth = 0.8, figsize = (10, 10), title =
'athletes.csv Medal dataset', grid = True, legend = True)
```

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e94eeb0>
```



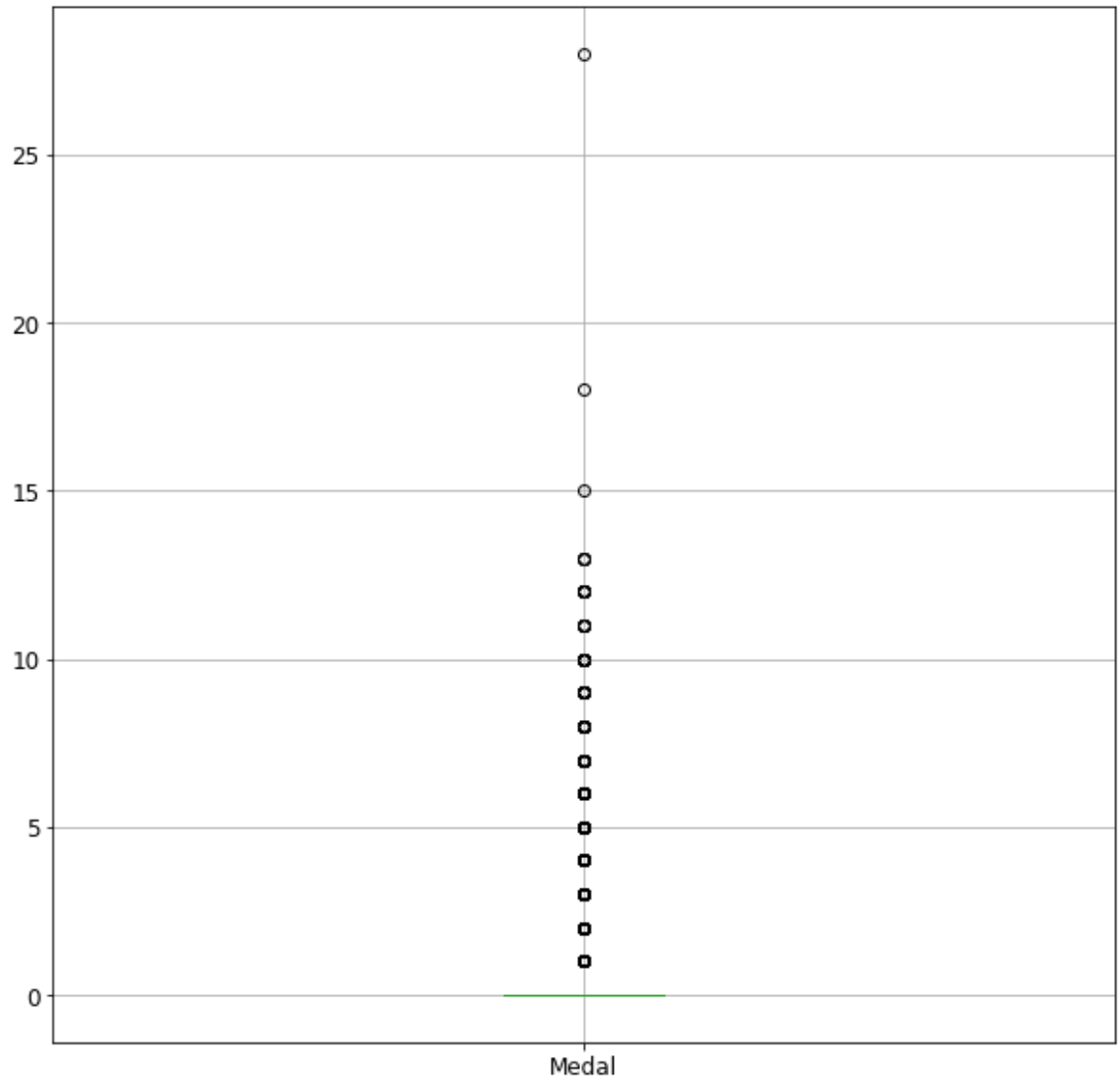
b. Box Plot

In [54]:

```
data5.boxplot(fontsize = 'large', figsize = (10, 10), grid = True)
```

Out[54]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e834b90>
```



2. Statistical Descriptions

a. Minimum Value

In [55]:

```
data5.min(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[55]:

```
' Medal  0'
```

b. Maximum Value

In [56]:

```
data5.max(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[56]:

```
' Medal  28'
```

c. Mean

In [57]:

```
data5.mean(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[57]:

```
' Medal  0.295'
```

d. Mode

In [58]:

```
data5.mode(numeric_only = True).round(3).to_string(index = None)
```

Out[58]:

```
' Medal\n     0'
```

e. Median

In [59]:

```
data5.median(numeric_only = True).round(3).reset_index().to_string(header = None, index
= None)
```

Out[59]:

```
' Medal  0.0'
```

f. Variance

In [60]:

```
data5.var(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[60]:

```
' Medal  0.525'
```

g. Standard Deviation

In [61]:

```
data5.std(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[61]:

```
' Medal  0.725'
```

h. Skewness

In [62]:

```
data5.skew(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[62]:

```
' Medal  4.82'
```

i. Kurtosis

In [63]:

```
data5.kurt(numeric_only = True).round(3).reset_index().to_string(header = None, index =
None)
```

Out[63]:

```
' Medal  51.563'
```

3. Distribution Function

Dari data kasus yang diajukan, dapat kami simpulkan bahwa persebaran data bersifat diskrit dan penggunaan fungsi distribusi yang cocok hanya fungsi distribusi bernoulli atau binomial dengan n = 1. Lalu pertanyaan yang diberikan hanya memiliki 2 kemungkinan yaitu antara salah atau benar tetapi hanya satu kondisi saja yang menentukan sukses atau tidaknya. Oleh karena itu kami memilih fungsi distribusi bernoulli yang bersifat diskrit.

4. Questions

a. Peluang Y meraih

i. tepat 0 medali

In [64]:

```
round((len(data5.loc[data5['Medal'] == 0]) / len(data5)), 3)
```

Out[64]:

```
0.791
```

ii. lebih dari 10 medali

In [65]:

```
round((len(data5.loc[data5['Medal'] > 10]) / len(data5)), 3)
```

Out[65]:

```
0.0
```

iii. tepat 3 medali

In [66]:

```
round((len(data5.loc[data5['Medal'] == 3]) / len(data5)), 3)
```

Out[66]:

```
0.01
```

iv. 1 atau 5 medali

In [67]:

```
round((len(data5.loc[(data5['Medal'] == 1) | (data5['Medal'] == 5)]) / len(data5)), 3)
```

Out[67]:

```
0.159
```

b. Confidence interval 95%

In [68]:

```
[(data5.mean(numeric_only = True) - 1.96 * data5.std(numeric_only = True) / math.sqrt(l
en(data5))).round(5).reset_index().to_string(header = None, index = None), (data5.mean(
numeric_only = True) + 1.96 * data5.std(numeric_only = True) / math.sqrt(len(data5))).r
ound(5).reset_index().to_string(header = None, index = None)]
```

Out[68]:

```
[' Medal  0.29141', ' Medal  0.29915']
```