In [3]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import scipy.stats as stats
from scipy.stats import linregress
from scipy.stats import norm
from matplotlib import pyplot as plt
```

In [39]:

```
np.random.seed(10)
population_ages1 = stats.poisson.rvs(loc = 18, mu = 35, size = 150000)
population_ages2 = stats.poisson.rvs(loc = 18, mu = 10, size = 100000)
populationage = np.concatenate((population_ages1, population_ages2))

print(population_ages1.mean())
```

```
53.00546
```

In [32]:

```
import random
sample_ages = np.random.choice(a = population_ages1, size = 500)

print(sample_ages.mean())
```

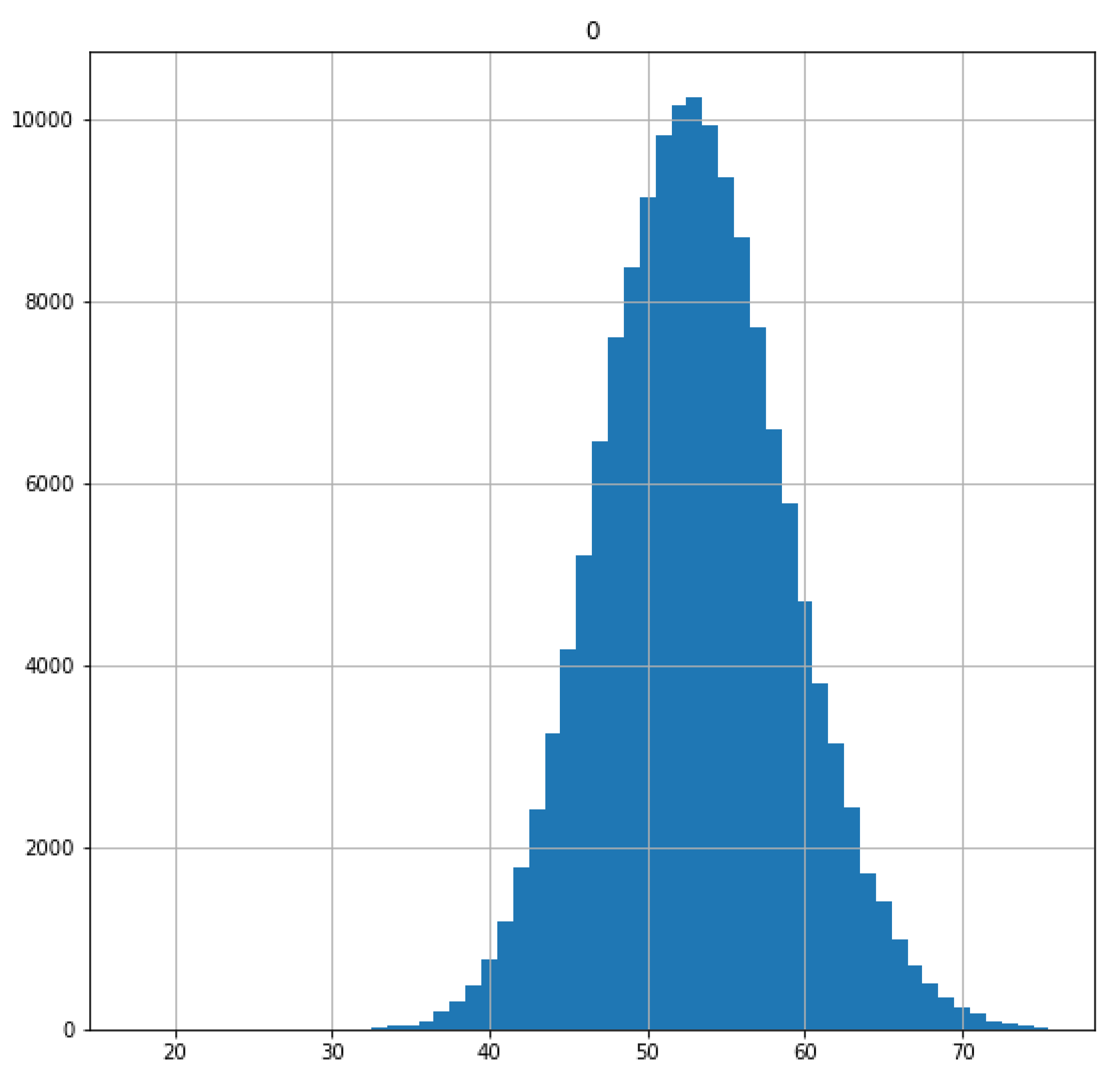
```
52.872
```

In [37]:

```
pd.DataFrame(population_ages1).hist(bins = 58, range = (17.5, 75.5), figsize = (9,9))
```

Out[37]:

```
array([[<AxesSubplot:title={'center':'0'}>]], dtype=object)
```



In [10]:

```
populationage.mean()
```

Out[10]:

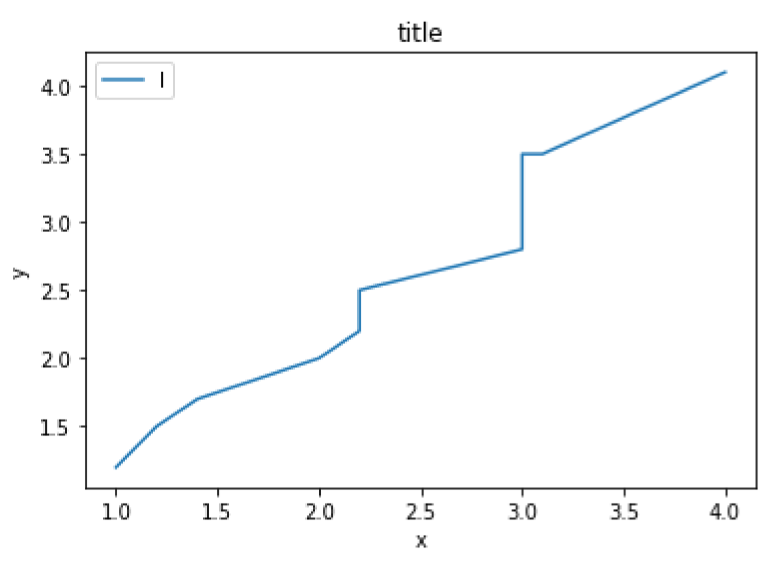
```
43.002372
```

In [13]:

```
x = np.array(sorted([1,2,3,4 ,1.2 ,2.2, 3, 2.2, 1.4, 3.1]))
y = np.array(sorted([1.5,2.2,3.5, 4.1, 1.2, 2.8, 2.0, 2.5, 1.7, 3.5]))
m, b = np.polyfit(x, y, 1)
h = (x,y)
plt.plot(x,y)
plt.title("title")
plt.xlabel("x")
plt.ylabel("y")
plt.legend("line")
plt.show
```

Out[13]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

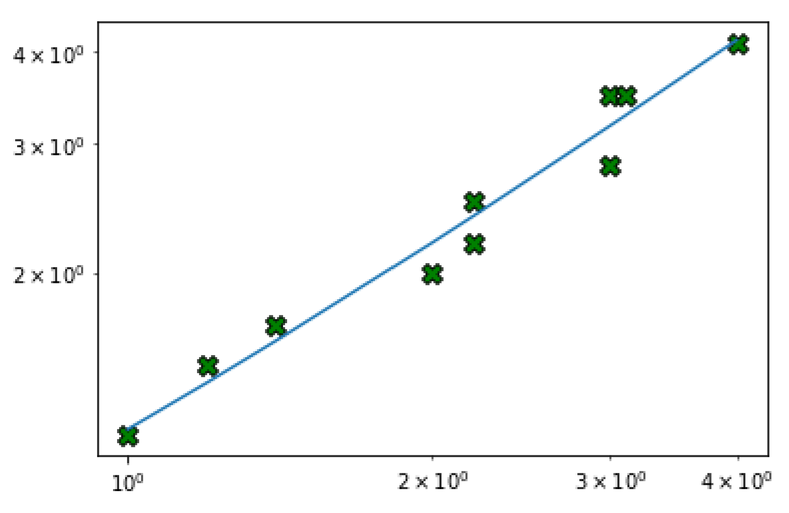


In [14]:

```
plt.scatter(x,y, s = 100, c = 'green', marker = "X", edgecolor = 'black', linewidth = 1)
plt.yscale('log')
plt.xscale('log')
plt.plot(x, m*x + b)
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x284718d64c0>]
```



In [ ]:

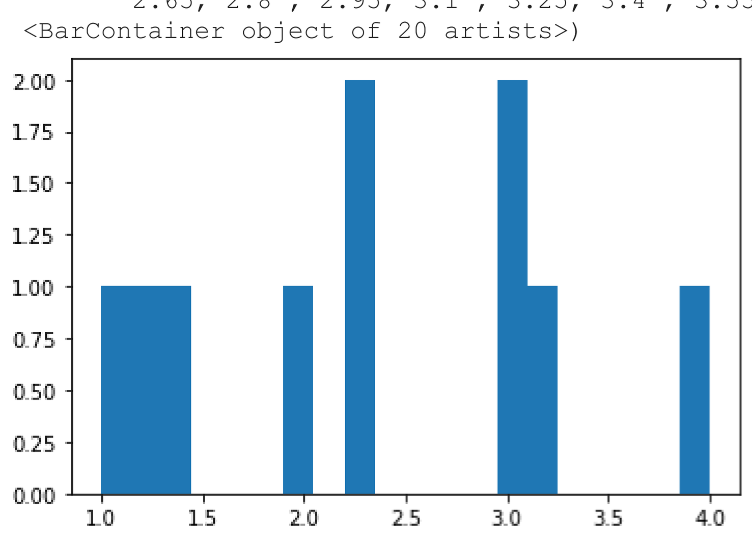
```
plt.savefig("blah.pdf")
```

In [50]:

```
plt.hist(x, 20)
```

Out[50]:

```
(array([1., 1., 1., 0., 0., 0., 1., 0., 2., 0., 0., 0., 0., 0., 2., 1., 0., 0.,
        0., 0., 1.]),
 array([1.  , 1.15, 1.3 , 1.45, 1.6 , 1.75, 1.9 , 2.05, 2.2 , 2.35, 2.5 ,
        2.65, 2.8 , 2.95, 3.1 , 3.25, 3.4 , 3.55, 3.7 , 3.85, 4.  ]),
 <BarContainer object of 20 artists>)
```



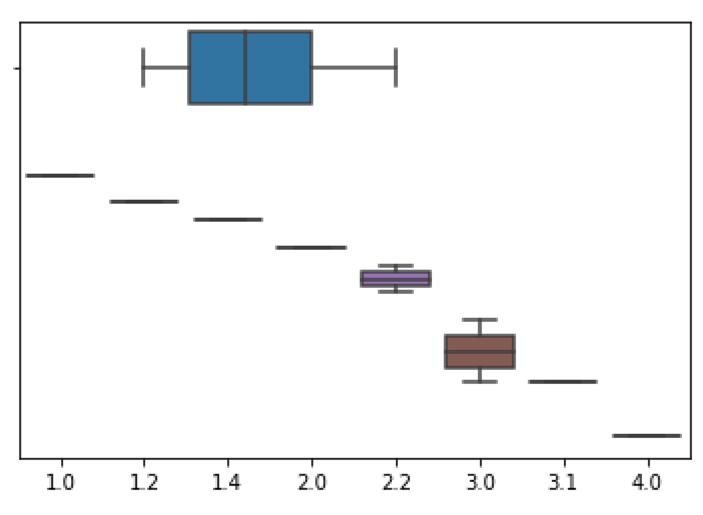
In [36]:

```
sns.boxplot(x)


sns.boxplot(x, y)
```

```
C:\Users\Home\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Home\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[36]:

```
<AxesSubplot:>
```

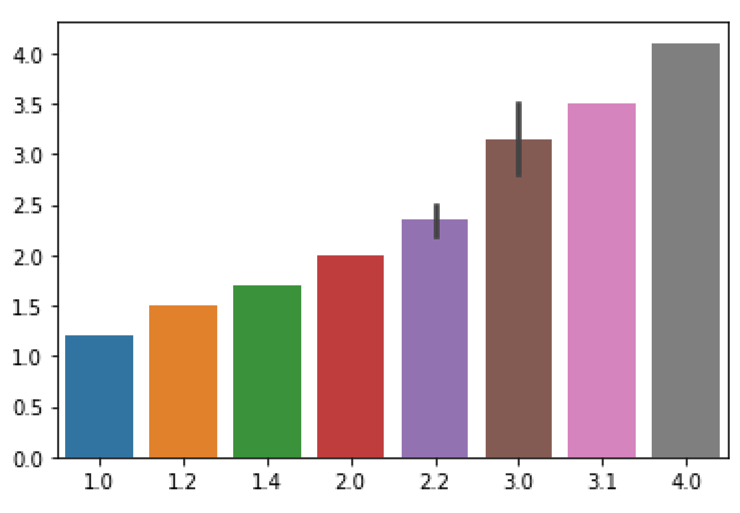


In [16]:

```
sns.barplot(x = x, y = y)
```

Out[16]:

```
<AxesSubplot:>
```

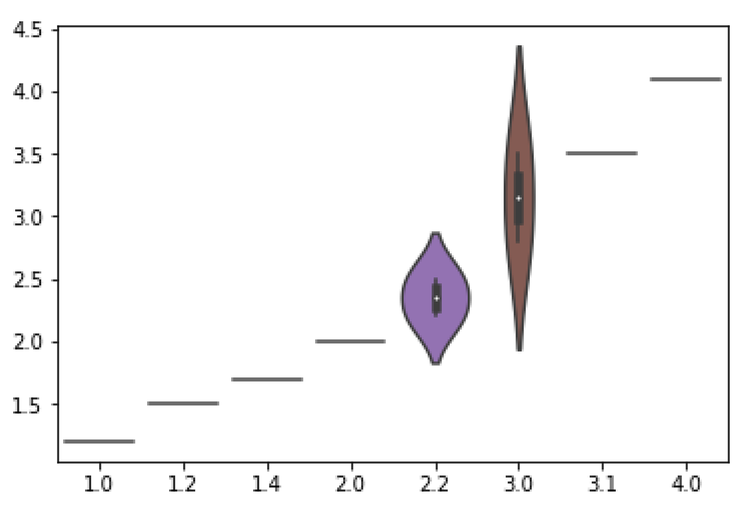


In [17]:

```
sns.violinplot(x=x, y=y)
```

Out[17]:

```
<AxesSubplot:>
```

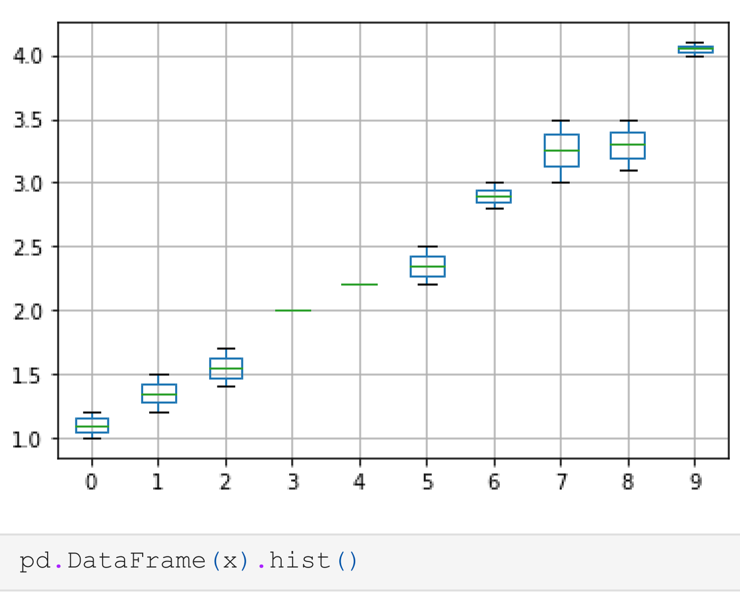


In [20]:

```
pd.DataFrame(h).boxplot()
```

Out[20]:

```
<AxesSubplot:>
```



In [16]:

```
pd.DataFrame(x).hist()
```

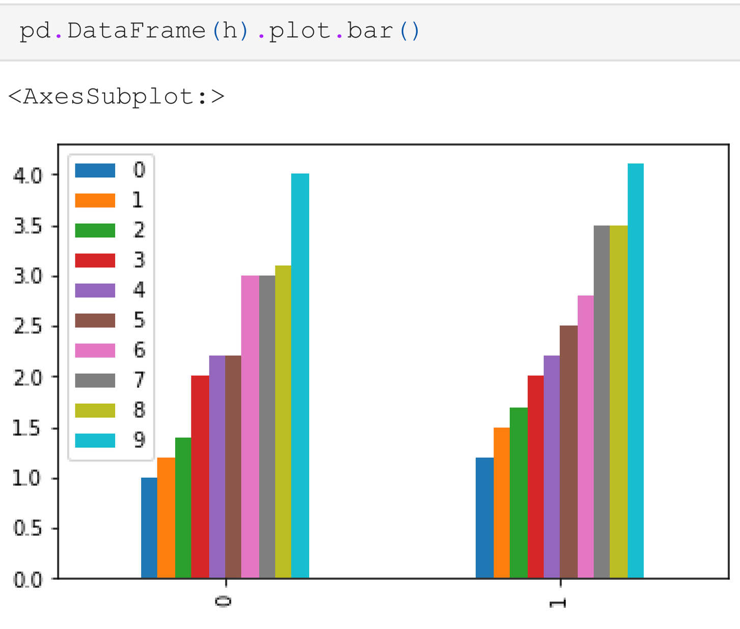
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-16-15b7d9c9a347> in <module>
----> 1 pd.h.hist()

~\anaconda3\lib\site-packages\pandas\__init__.py in __getattr__(name)
    256                 return _SparseArray
    257 
--> 258         raise AttributeError(f"module 'pandas' has no attribute '{name}'")
    259 
    260 

AttributeError: module 'pandas' has no attribute 'h'
```

In [19]:

```
pd.DataFrame(h).plot.bar()
```

Out[19]:

```
<AxesSubplot:>
```

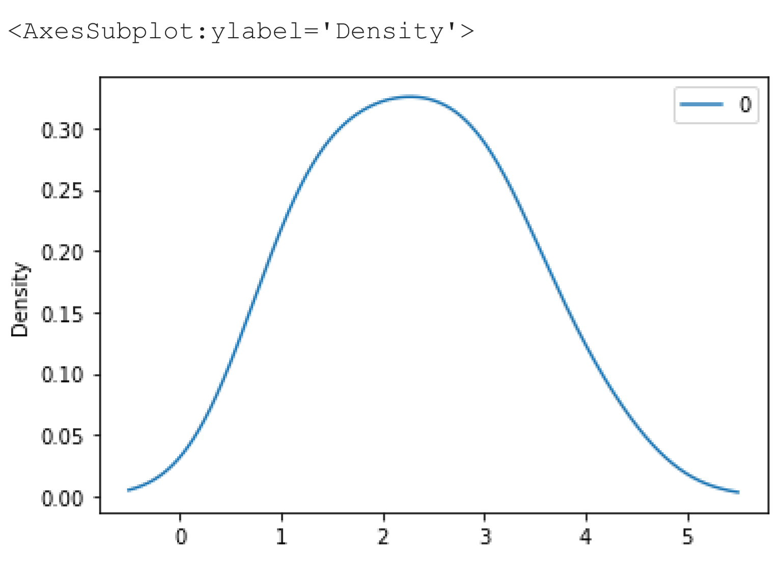


In [21]:

```
pd.DataFrame(x).plot.kde()
```

Out[21]:

```
<AxesSubplot:ylabel='Density'>
```



In [23]:

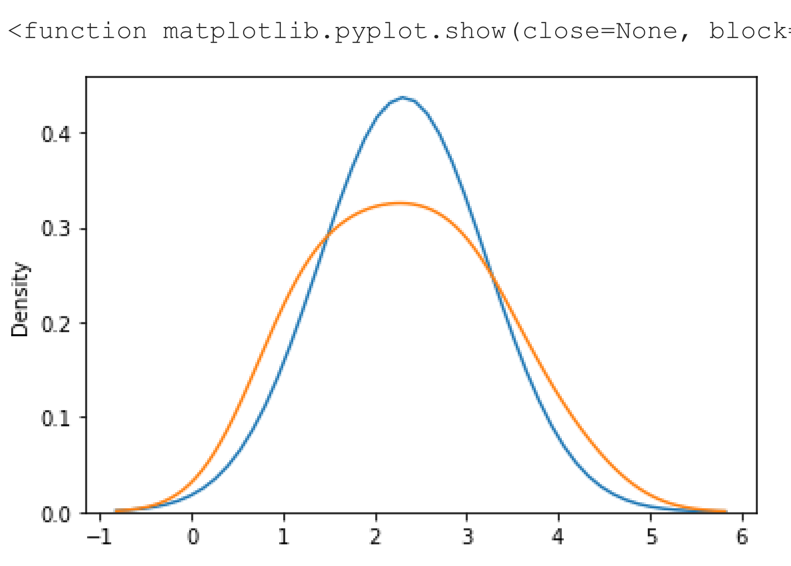
```
dist = norm(x.mean(), x.std())
```

In [31]:

```
xs = np.linspace(x.min()-2*x.std(), x.max()+2*x.std())
ys = dist.pdf(xs)
plt.plot(xs, ys)
sns.kdeplot(x)
plt.show
```

Out[31]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [30]:

```
linregress(xs, ys)
```

Out[30]:

```
LinregressResult(slope=-0.007438996261114978, intercept=0.16703865744547955, rvalue=-0.0936058178615684, pvalue=0.5179077553060198, stderr=0.011420361207846218)
```

In [ ]:

```
dataset['column'].replace([value], np.nan, inplace=True)
```