

```
In [1]: 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 [2]: cities = pd.read_csv('C:/Users/Home/Desktop/bigcity.csv')
```

```
In [4]: cities.head(7)
cities.tail(2)
```



```
In [5]: cities.columns
```

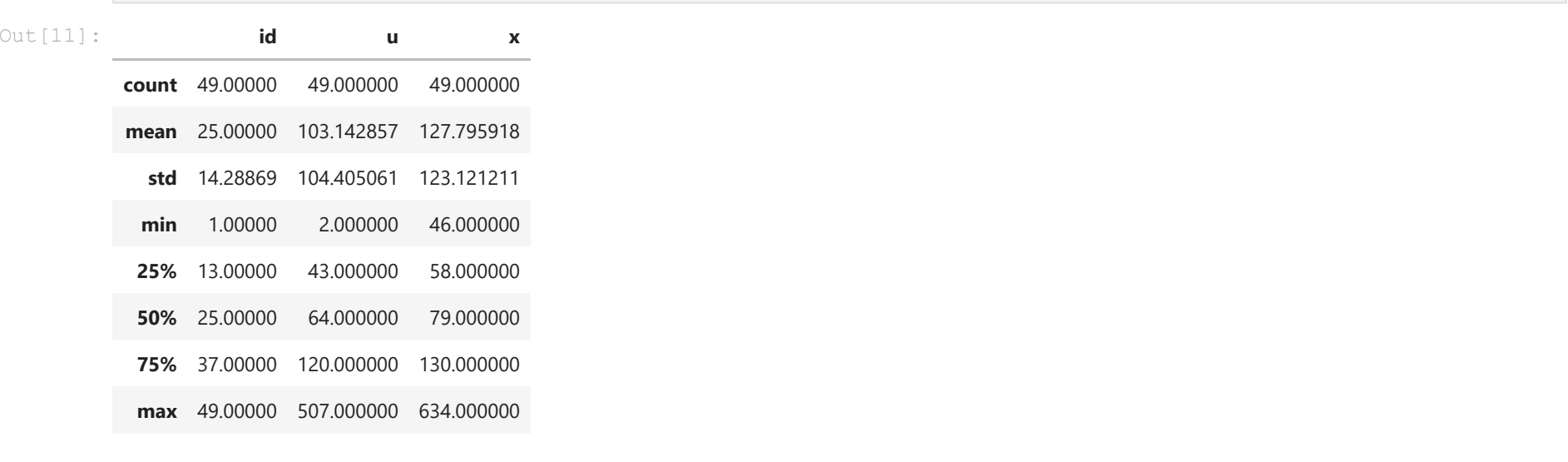
```
Out[5]: Index(['Unnamed: 0', 'u', 'x'], dtype='object')
```

```
In [51]: cities = cities.drop(columns = "id")
```

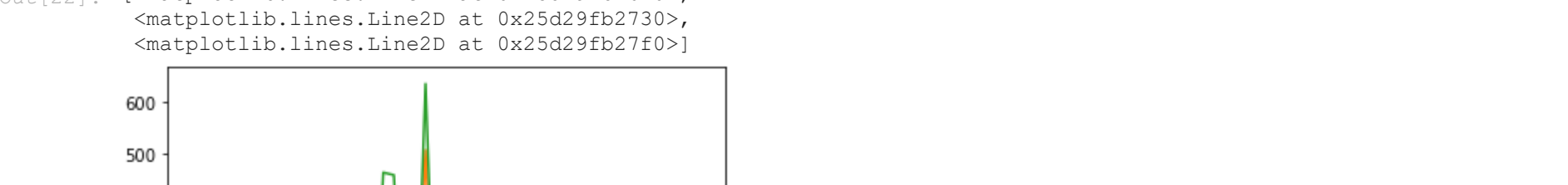
```
In [10]: cities.shape
```

```
Out[10]: (49, 3)
```

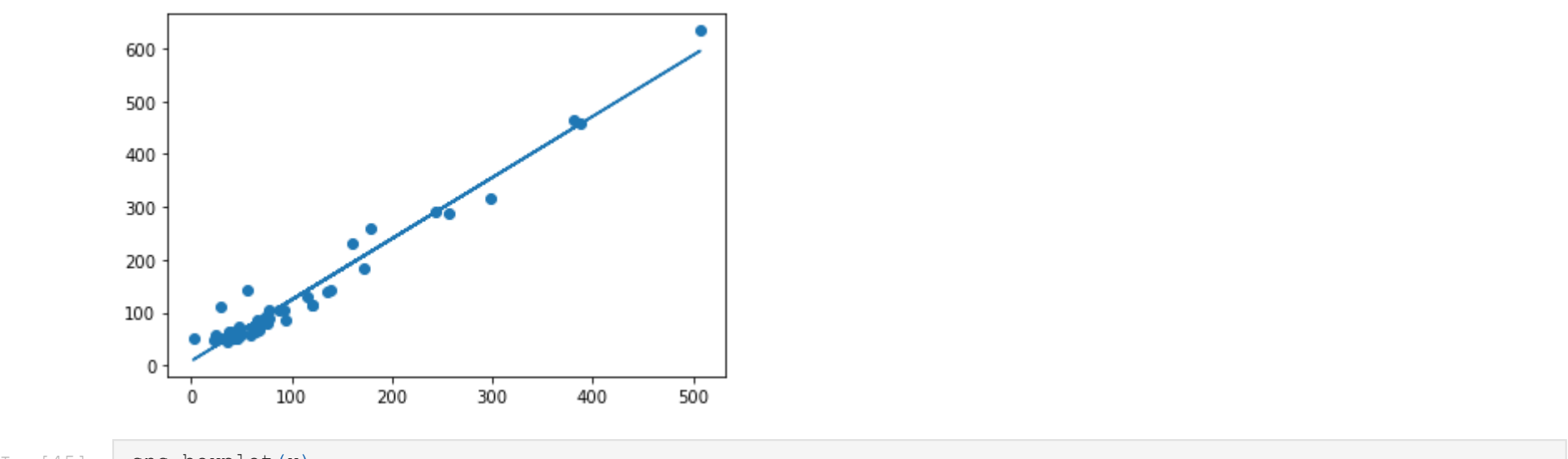
```
In [11]: cities.describe()
```



```
In [22]: plt.plot(cities)
```



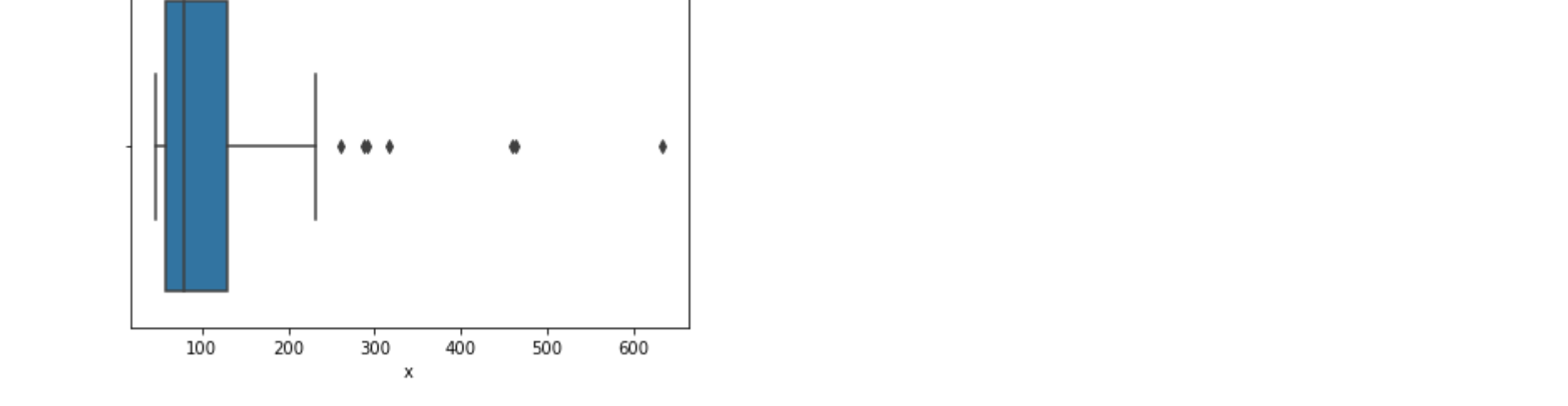
```
In [26]: x = cities['x']
u = cities['u']
plt.scatter(u, x)
m, b = np.polyfit(u, x, 1)
plt.plot(u, m*u + b)
```



```
In [45]: sns.boxplot(x)
```

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.

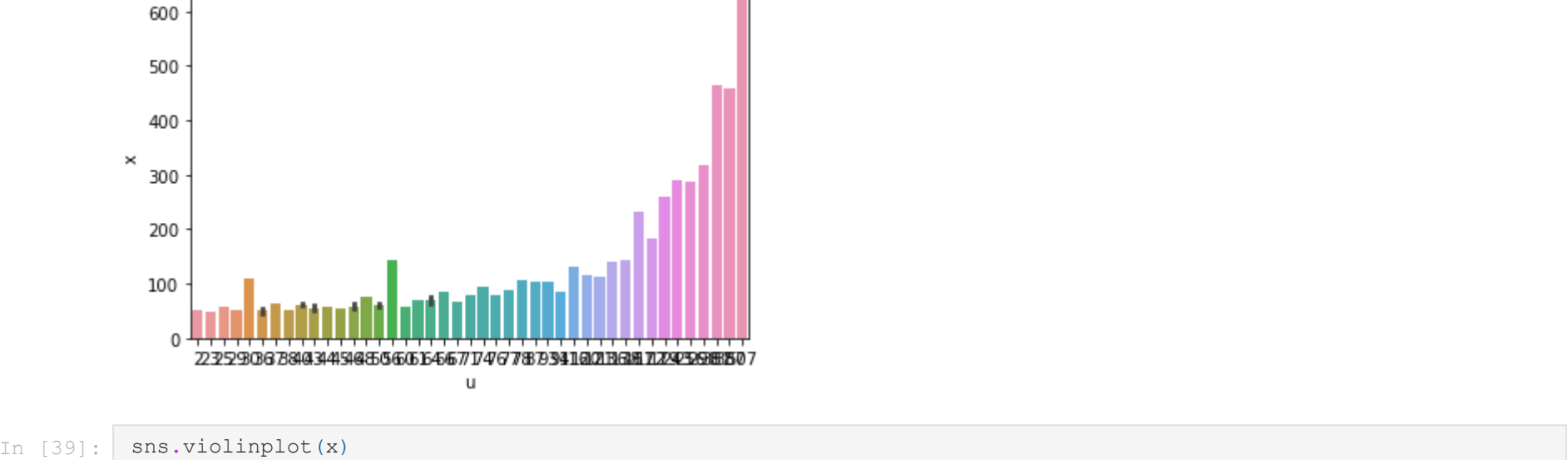
warnings.warn()



```
In [38]: sns.barplot(u,x)
```

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.

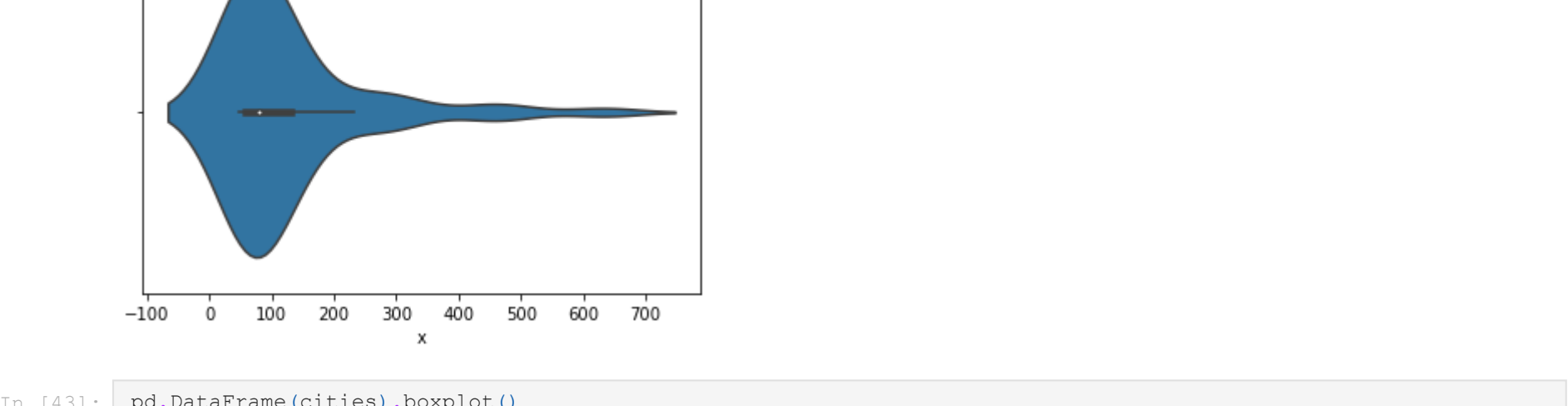
warnings.warn()



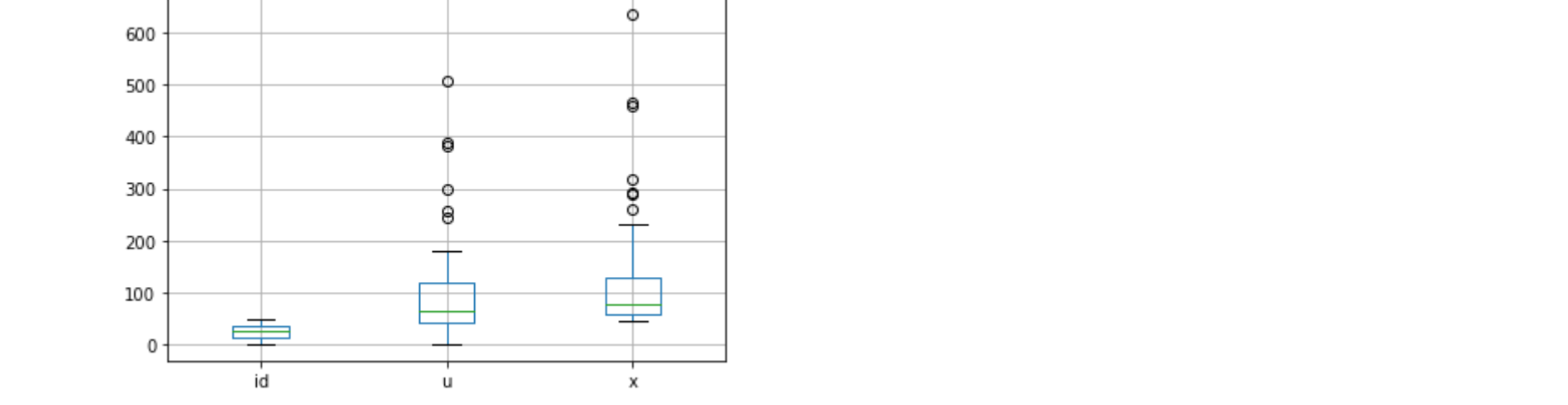
```
In [39]: sns.violinplot(x)
```

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.

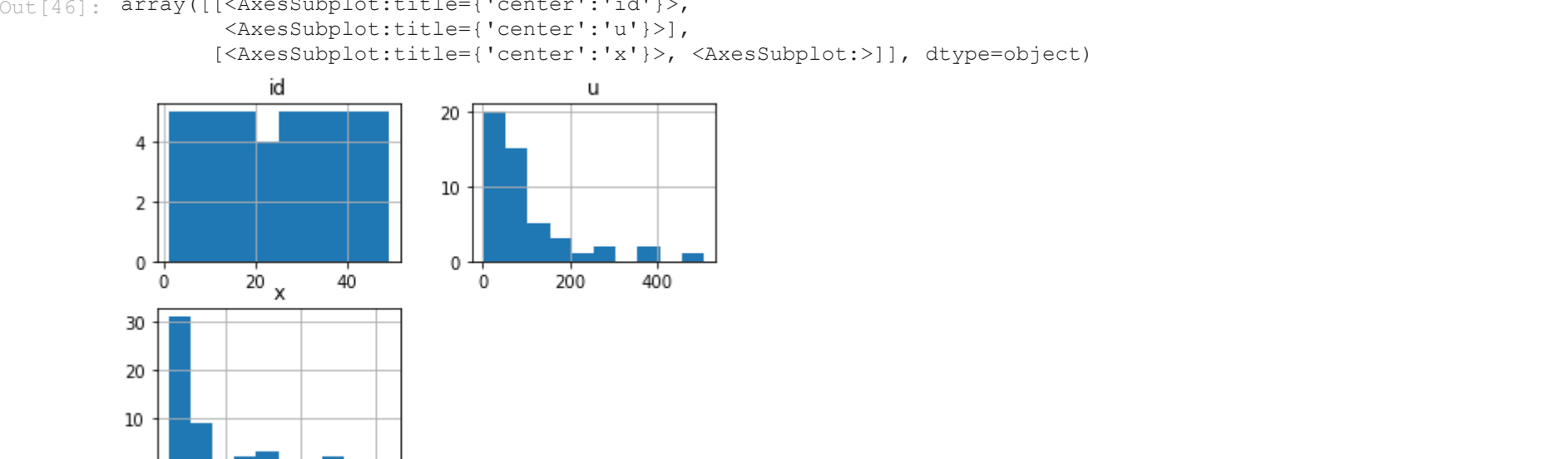
warnings.warn()



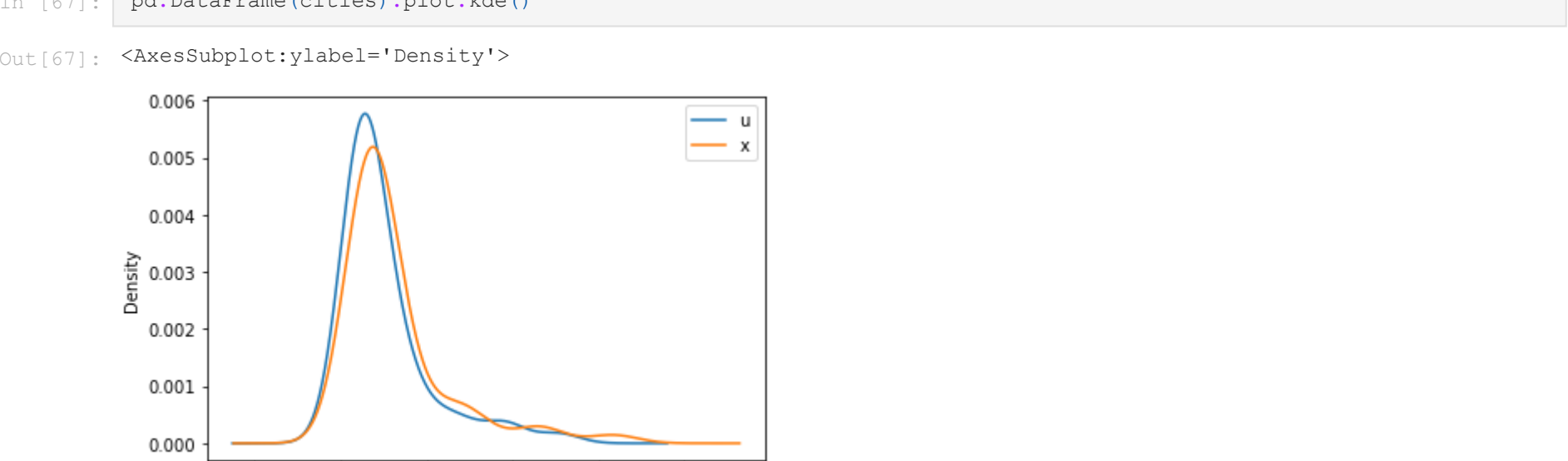
```
In [43]: pd.DataFrame(cities).boxplot()
```



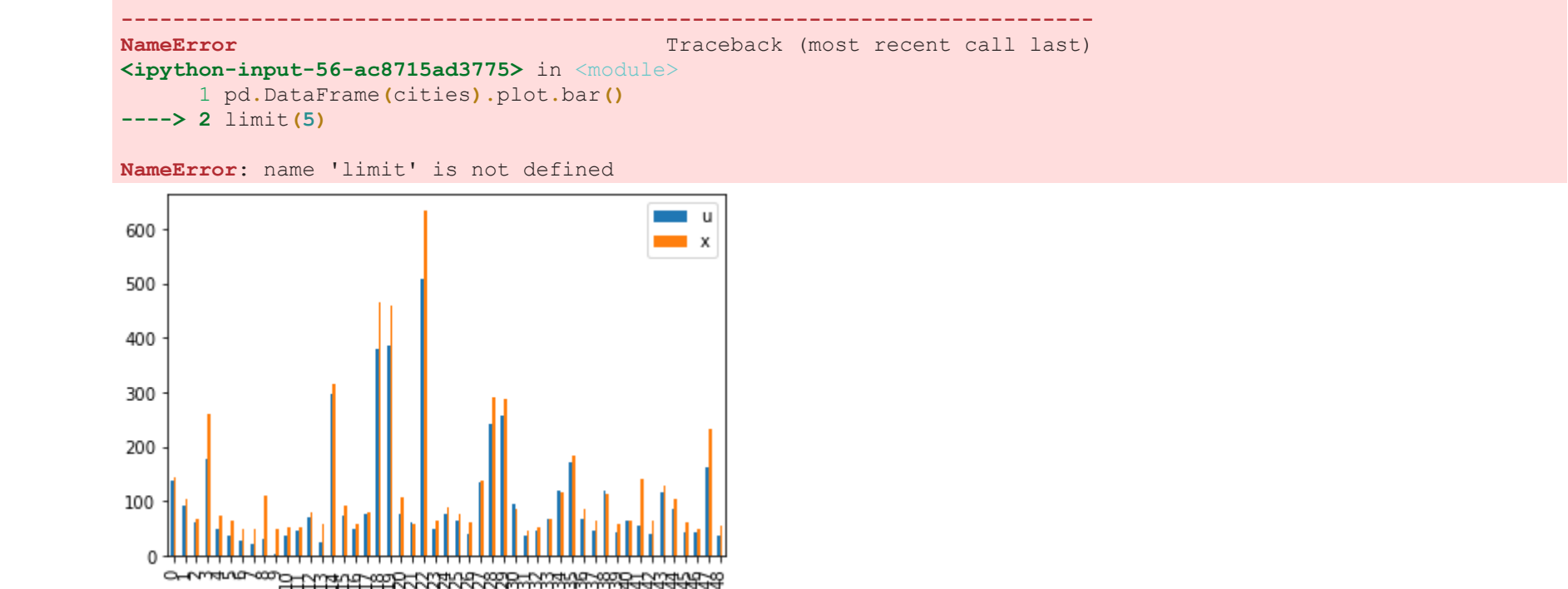
```
In [46]: pd.DataFrame(cities).hist()
```



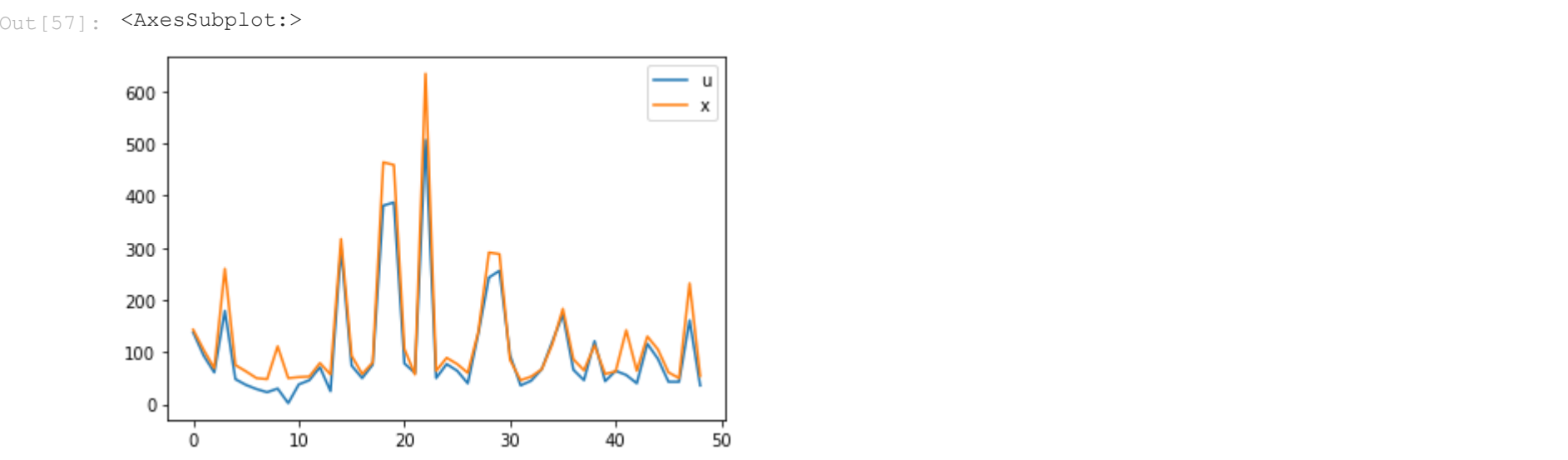
```
In [67]: pd.DataFrame(cities).plot.kde()
```



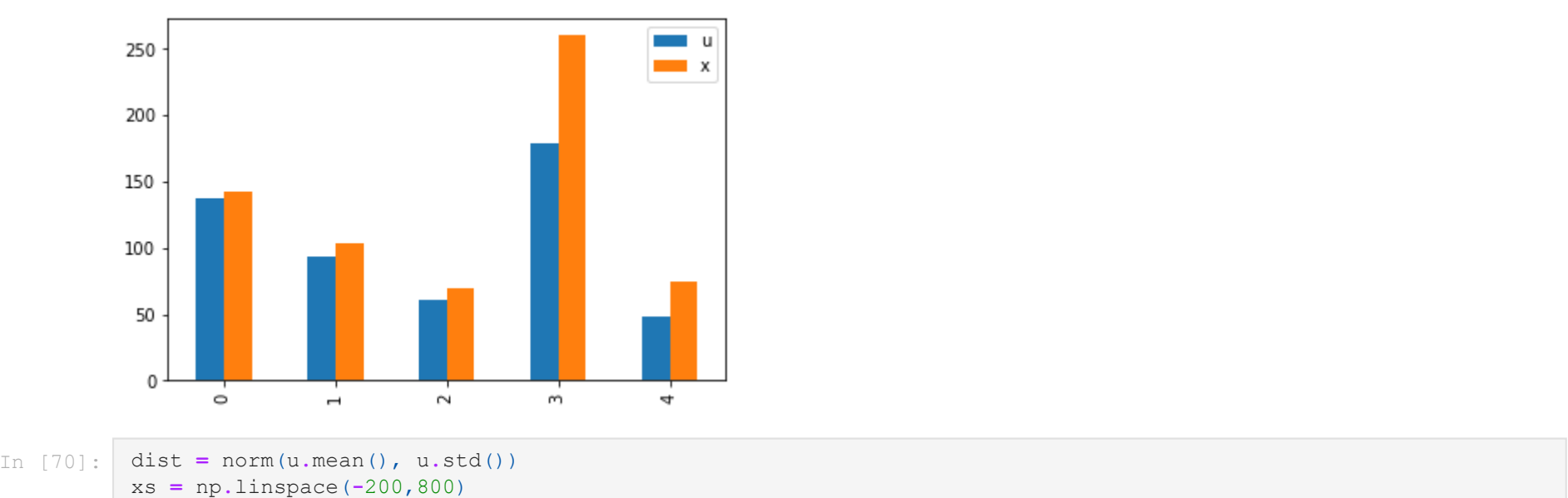
```
In [56]: pd.DataFrame(cities).plot.bar()
head
```



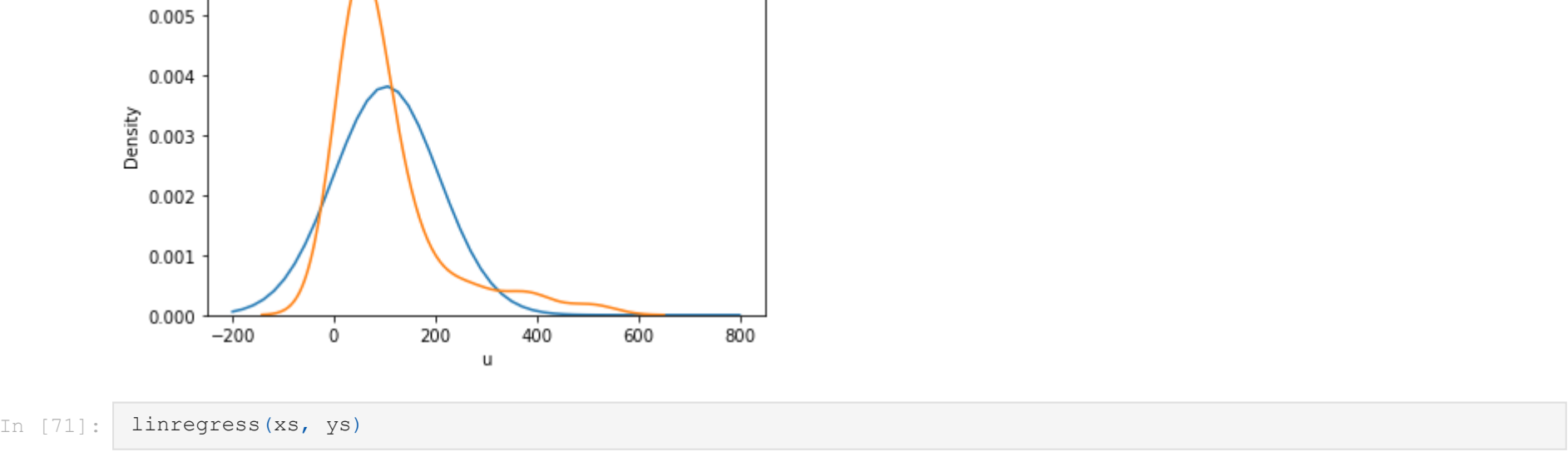
```
In [57]: cities.plot.line()
```



```
In [64]: g = cities.head(5)
g.plot.bar()
```



```
In [70]: dist = norm(u.mean(), u.std())
xs = np.linspace(-200,800)
ys = dist.pdf(xs)
plt.clf()
plt.plot(xs, ys)
sns.kdeplot(u)
```



```
In [71]: linregress(xs, ys)
```

```
Out[71]: LinregressResult(slope=-2.2161684554950963e-06, intercept=0.0016435336429545683, rvalue=-0.5020601352474419, pval=0.00020332570688773712, stderr=5.510087811430419e-07)
```

```
In [87]: cities['x'].replace([48], 100, inplace = True)
```

```
In [88]: cities.min()
```

