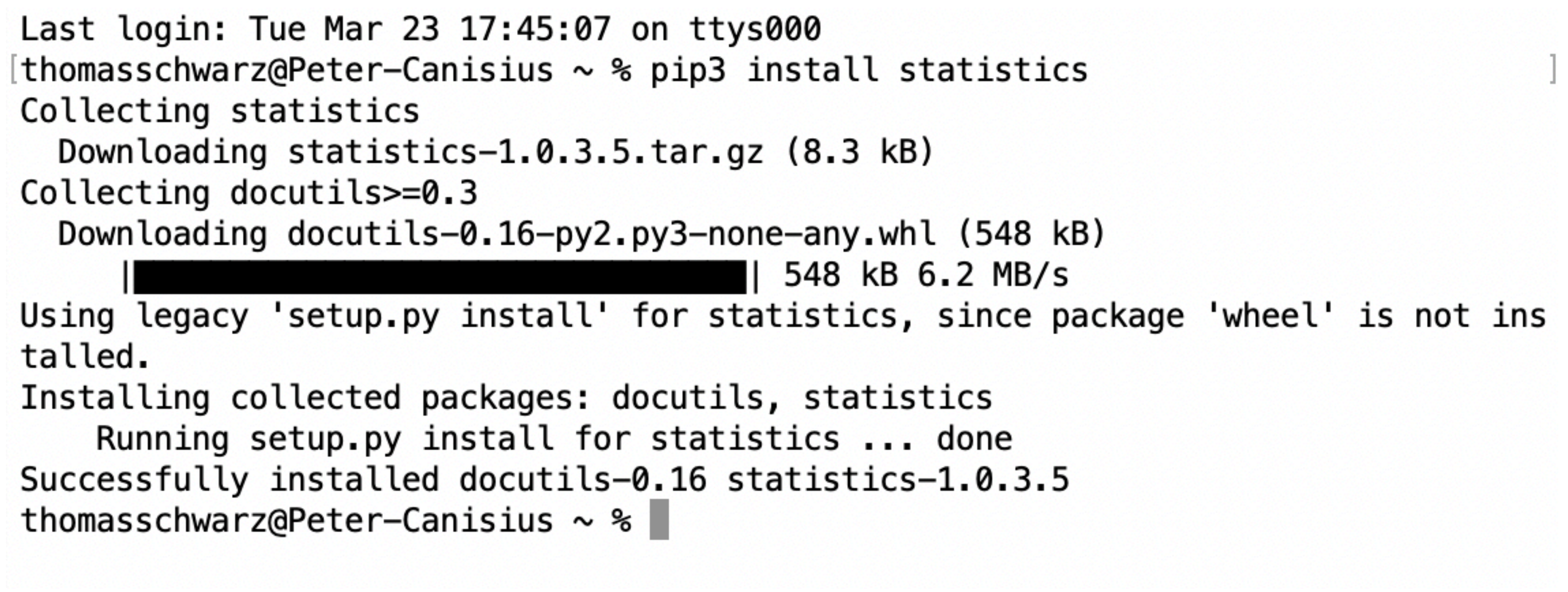
Descriptive Statistics with Python

Thomas Schwarz, SJ

Statistics Modules

- Python has a number of statistics modules
    - A simple one is called statistics
    - Use pip3 install statistics



```
Last login: Tue Mar 23 17:45:07 on ttys000
thomasschwarz@Peter-Canisius ~ % pip3 install statistics
Collecting statistics
  Downloading statistics-1.0.3.5.tar.gz (8.3 kB)
Collecting docutils>=0.3
  Downloading docutils-0.16-py2.py3-none-any.whl (548 kB)
     |████████████████████████████████| 548 kB 6.2 MB/s
Using legacy 'setup.py install' for statistics, since package 'wheel' is not ins
talled.
Installing collected packages: docutils, statistics
    Running setup.py install for statistics ... done
Successfully installed docutils-0.16 statistics-1.0.3.5
thomasschwarz@Peter-Canisius ~ %
```

Example

- Population of state capitals
- Use a weird file available at
    - https://tschwarz.mscs.mu.edu/Classes/COSC1000/Modules/capitals.csv
    - Made available from someone who took it from Wikipedia

Example

- Printing out line for line
  - We find encoding errors
  - So we that the encoding to 'latin'

```
with open('capitals.csv', encoding='latin') as infile:
    for line in infile:
        print(line)
```

Example

- Two title lines
- Then data: But notice the string literals with commata

```
1,Alabama,AL,1819,Montgomery,1846,155.4,No,"2,05,764","3,74,536"
,Birmingham is the state's largest city
```

- So, it is a comma-separated file with commas in strings
- Instead of writing our own interpreter, we split the file along "
    - This works because we only want to extract the population number

Example

```
1,Alabama,AL,1819,Montgomery,1846,155.4,No,"2,05,764","3,74,536"
,Birmingham is the state's largest city
```

- We are interested in getting the second value

```
with open('capitals.csv', encoding='latin') as infile:
    infile.readline()
    infile.readline()
    for line in infile:
        values = line.split('"')
        print(values)
```

```
['1,Alabama,AL,1819,Montgomery,1846,155.4,No,', '2,05,764', ',',
'3,74,536', ",Birmingham is the state's largest city\n"]
```

Example

```
with open('capitals.csv', encoding='latin') as infile:
    infile.readline()
    infile.readline()
    for line in infile:
        values = line.split('"')
        print(values[1])
```

```
2,05,764
31,275
14,45,632
1,93,524
4,66,488
...
```

- A weird Indian looking format

Example

- Creating our own function to remove commata

```
def remove(line, symbol):
    result = [ ]
    for letter in line:
        if letter != symbol:
            result.append(letter)
    return ''.join(result)
```

Example

- Now we place the numbers into an array

```
pops = [ ]
with open('capitals.csv', encoding='latin') as infile:
    infile.readline()
    infile.readline()
    for line in infile:
        value = remove(line.strip().split('"')[1],',')
        pops.append(int(value))
```

Example

- Finally, we can apply statistics

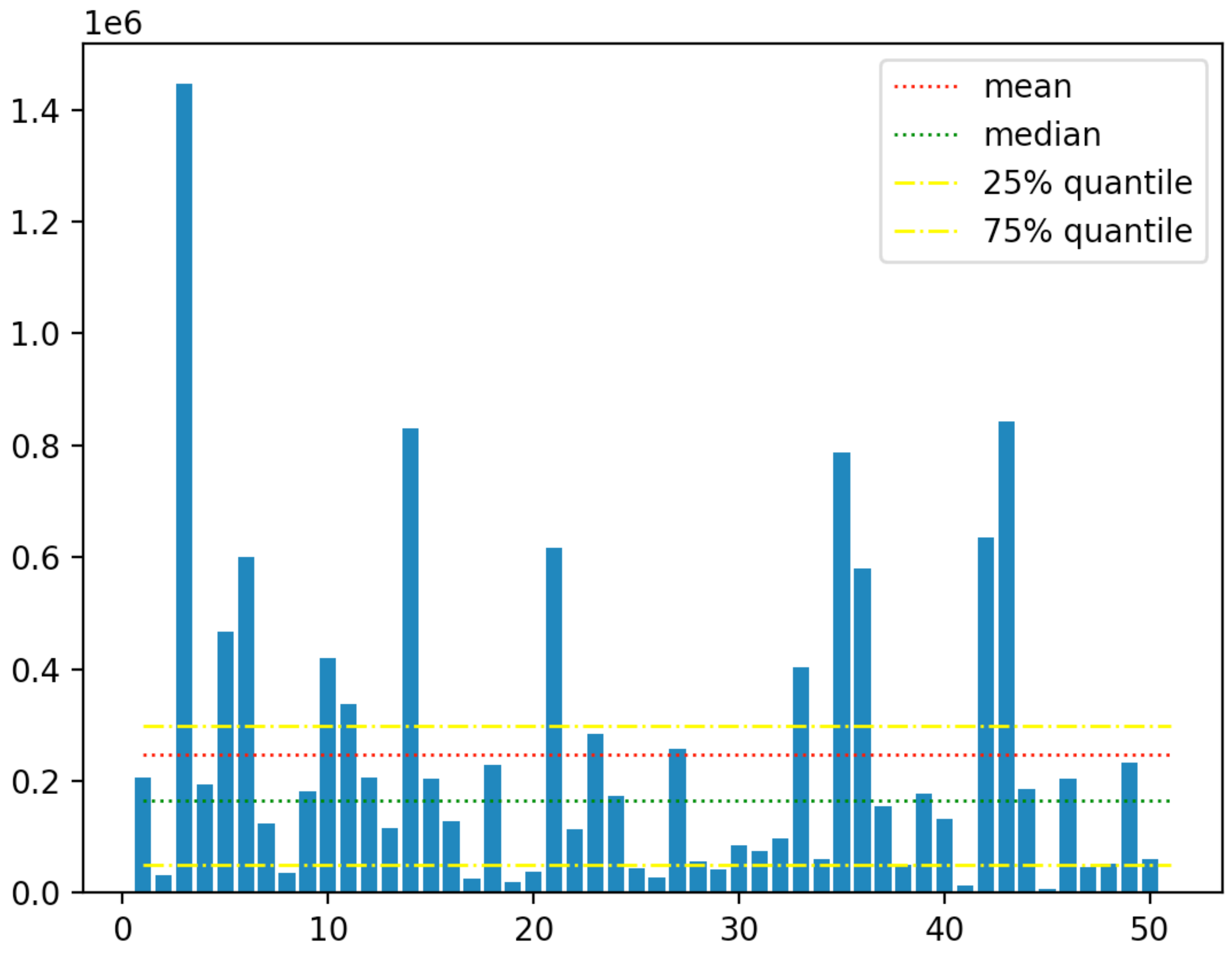
```
print('mean', stats.mean(pops))
print('median', stats.median(pops))
print('pstdev', stats.pstdev(pops))
print('quantiles', stats.quantiles(pops))
```

Statistics

- Descriptive Statistics:
  - Mean (arithmetic mean)
  - Median (half the values above, half the values below)
  - Quantiles
    - 25% below, 75% above
    - 50% below, 50% above
    - 75% below, 25% above
  - Standard Deviation: Measure for the average distance of a point from the mean

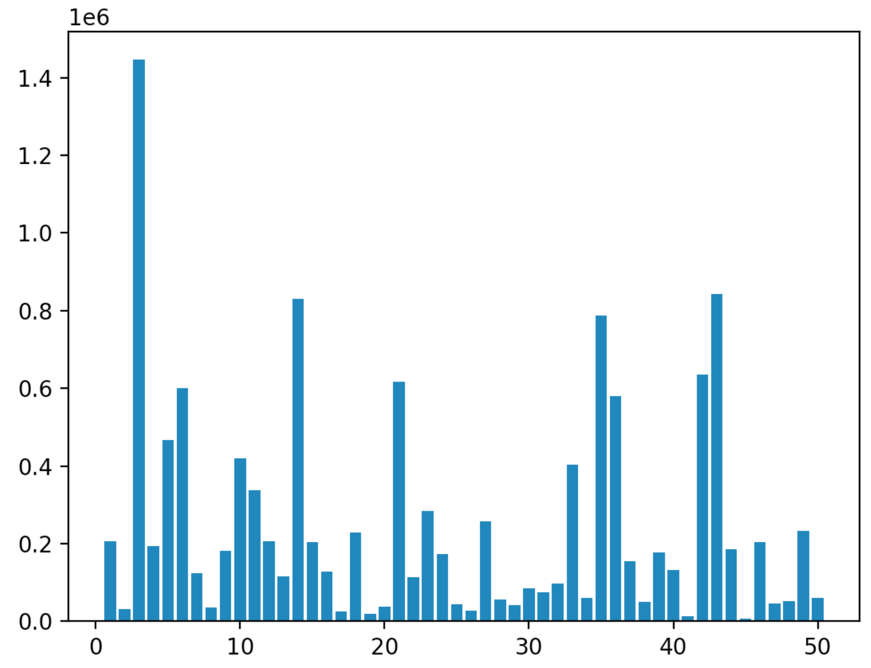
Statistics

- Median is liked because it is less sensitive to outliers
- Quantiles and standard deviation help with visualizing distrik



Example: Visualization

- We want to present the values:
  - Use a bar chart
    - Needs X and Y values
      - X numbers between 1 and 50
      - Y population numbers
      - `plt.bar(range(1,51), pops)`

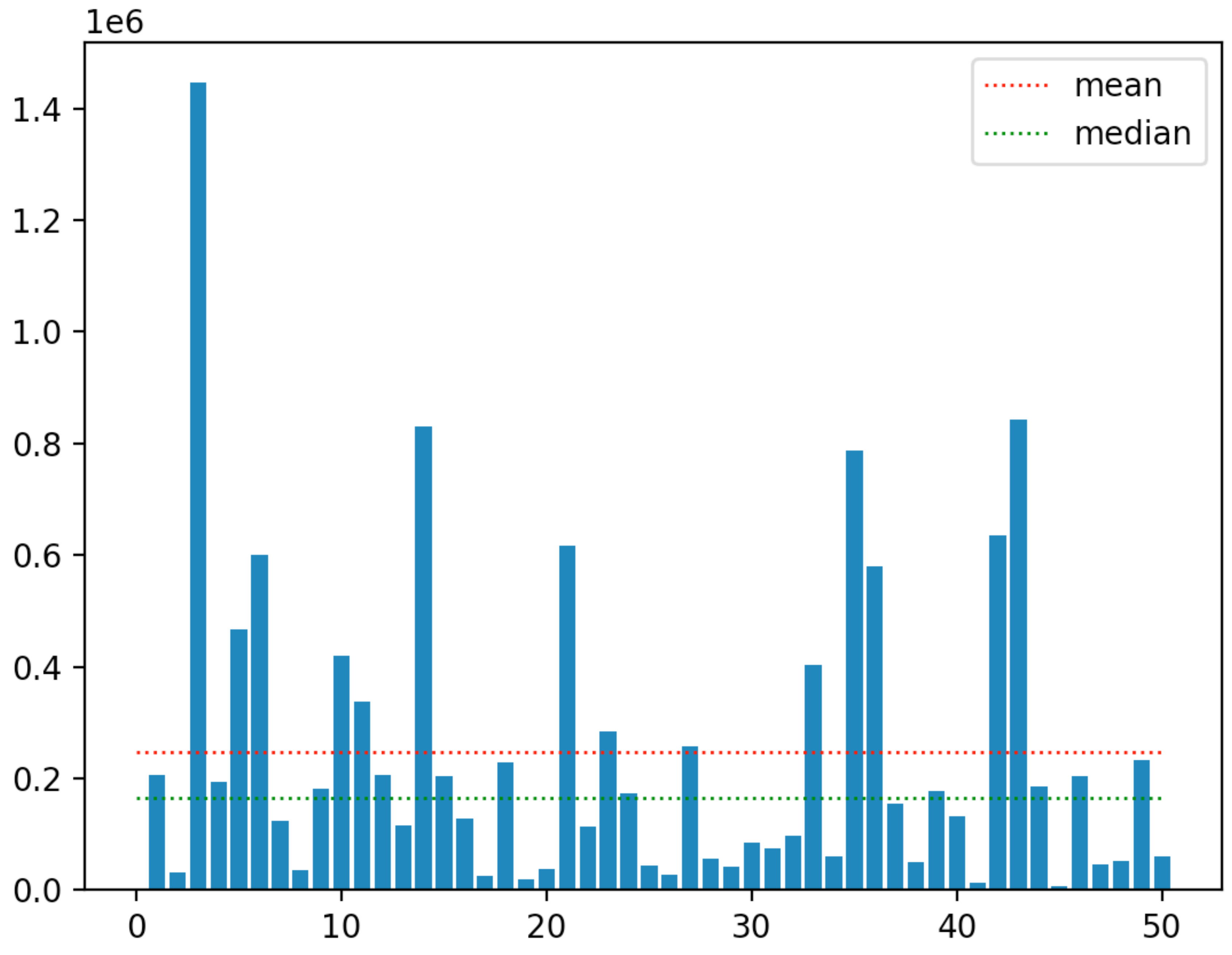


Example

- To draw the mean and median:
  - Plot a line
  - Using abbreviations

```
plt.bar(range(1,51), pops)
plt.plot([0,50], 2*[stats.mean(pops)], lw='1', ls=':',
label='mean', c='red')
plt.plot([0,50], 2*[stats.median(pops)], lw='1', ls=':',
label='median', c='green')
plt.legend()
plt.show()
```

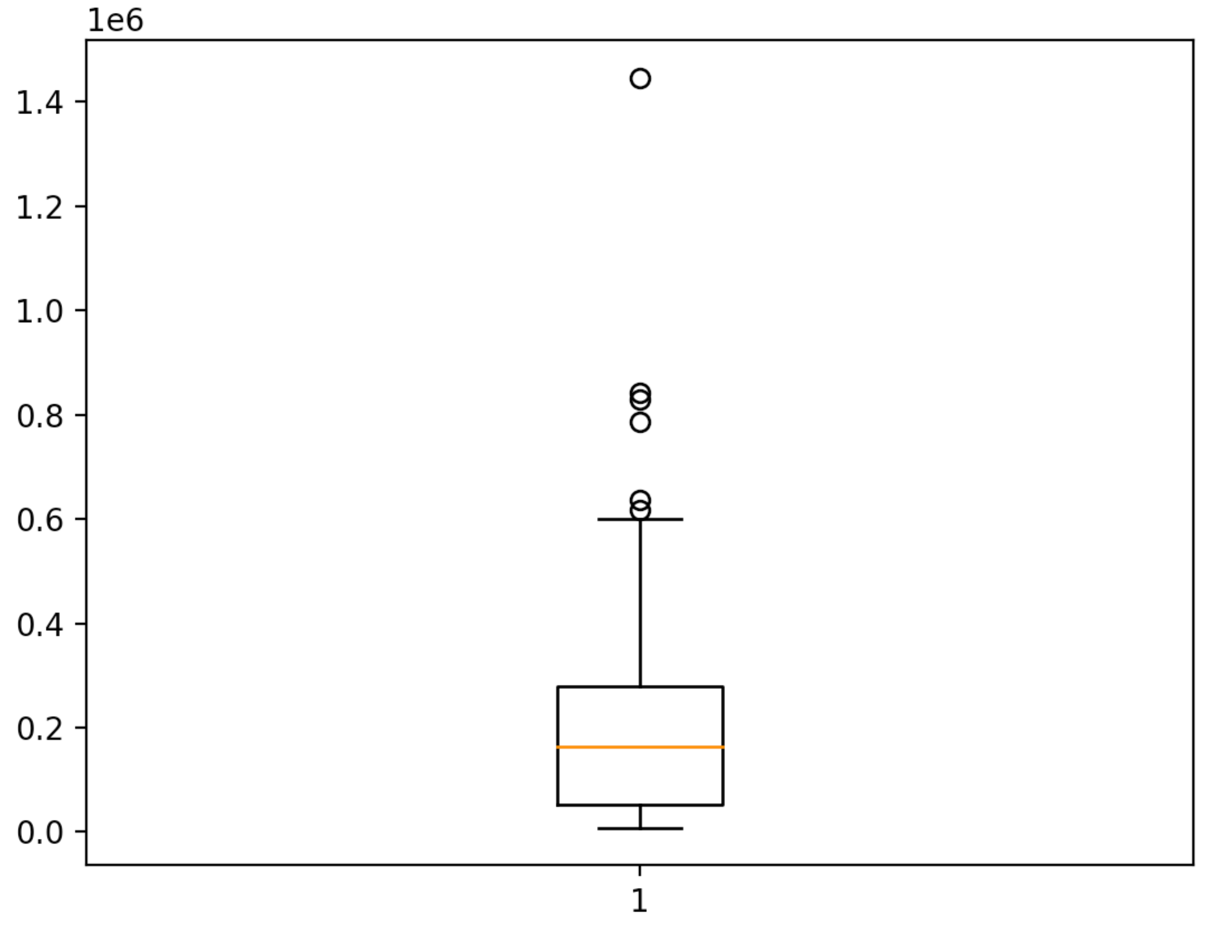
Example



Example

- Box and whisker plot: Box extends from lower to upper quartile with line for median
- Whiskers show range, with "fliers" (outliers) above and below

```
plt.boxplot(pops)
plt.show()
```


1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
1