q2

February 10, 2021

[1]:
```
%matplotlib inline

from numpy import mean, std, array, median, percentile
from numpy.random import random
from matplotlib.pyplot import boxplot
from math import sqrt
```

[2]:
```
x = random((1000,))
# X
```

Part A: Compute the sample's Mean and Standard Deviation Using Numpy functions

[3]:
```
sample_mean_x = mean(x)
sample_mean_x
```

[3]:
```
0.5057000380848997
```

[4]:
```
sample_std_x = std(x)
sample_std_x
```

[4]:
```
0.29539677637048345
```

Part B: Compute the standardized version of the original sample of x. We can convert the dataset to a standard normal distribution by applying the following standardization to each x_i

$$z = \frac{x - \bar{x}}{\sigma}$$

[5]:
```
def standardize(x, sample_mean, std):

    x_standardized = (x - sample_mean) / std

    return x_standardized
```

[6]:
```
x_standardized = array([standardize(x_i, sample_mean_x, sample_std_x) for x_i␣
↪in x])
# x_standardized
```

We took a normal distribution and standardized it so that it becomes a standard normal distribution. By definition of the standard normal distribution, the mean should be 0 and the standard deviation should be 1. We check in the code below:

[7]:
```
mean_standardized = mean(x_standardized)
mean_standardized
```

[7]: 1.8474111129762604e-16

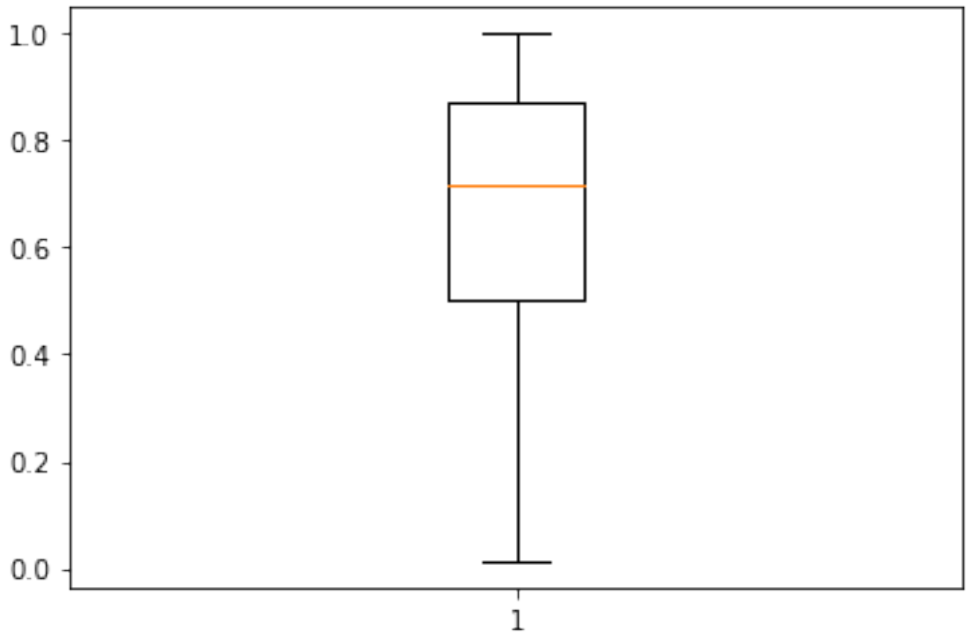
[8]:
```
std_standardized = std(x_standardized)
std_standardized
```

[8]: 1.0

Part C: Draw a boxplot of the set of square roots of the elements of the sample x (plot with matplotlib.pyplot.boxplot)

[9]:
```
x_square_roots = array([sqrt(u) for u in x])

box = boxplot(x_square_roots)
# box
```



Part D: Using Numpy, compute the numerical locations of the three parallel line segments that form the main body of the box plot above

In other words, we need to find the 3rd quartile (topmost line), the 1st quartile (bottommost line), and the median (the line in the middle)

[10]:
```
# middlemost line -> median

median = median(x_square_roots)
median
```

[10]: 0.7152223829609285

[11]:
```
# topmost line -> 3rd quartile
# the 3rd quartile is the same as the 75th percentile

quartile3 = percentile(x_square_roots, 75)
quartile3
```

[11]: 0.8701397127134112

[12]:
```
# bottommost line -> 1st quartile
# the 1st quartile is the same as the 25th percentile

quartile3 = percentile(x_square_roots, 25)
quartile3
```

[12]: 0.5030572095323681