Congratulations! You passed!

Grade received 100%

To pass 100% or higher

Go to next item

Generating Random Data and Samples

Latest Submission Grade 100%

1. In the code block below, generate 3 normal random variables with mean 100 and standard deviation 1.

1 / 1 p

This will require about 4 lines of code. Use the functions provided in this outline.

- Import the *numpy* library
- Set the seed to 123 to initialize environment so random variables are replicated according to the grader. (*hint: np.random.seed(**?**)*)
- Generate three random normal variables with mean 100 and standard deviation 1 and assign them to a variable named sample. (*hint: sample = np.random.normal(**?**,**?**,**?**)*)
- Print the variable sample.

*The **question marks** in the hints indicate input parameters.*

Choose the answer that matches your result to three decimal places.

Reference Documentation

- https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.random.seed.html
- https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.random.normal.html

```
# Write your function here
mean = 100
sd = 1

import numpy as np
np.random.seed(123)
variables = np.random.normal(mean, sd, 3)
print("Variables : ", variables)
```

Run

Reset

○ 99.914 101.937 100.282

○ 99.922 100.103 100.819

● 98.914 100.997 100.283

○ 99.822 100.093 100.719

○ 100.915 99.997 101.283

✓ Correct

2. Generating random samples from a population lies at the heart of statistics. In the code block below, draw a sample of size 10 from a set containing the integers 1 through 100.

1 / 1 p

This will require about 5 lines of code. Use the functions provided in this outline.

1. Import the numpy library
2. Set the seed to 123 to initialize environment so random variables are replicated according to the grader. (*hint: np.random.seed(**?**)*)
3. Create a vector called population, and put the numbers 1-100 into the population list. (*hint: np.arange(**?**,**?**)*)
4. Generate a sample with length 10 from the population. (*hint: sample = np.random.choice(**?**, **?**)*) and assign the output to a variable named sample.
5. Print the variable sample.

The **question marks** in the hints above indicate input parameters.

Reference Documentation

- https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.random.seed.html
- https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.arange.html
- https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.random.choice.html

```
import numpy as np
np.random.seed(123)
#https://numpy.org/doc/stable/reference/generated/numpy.arange.html
#up to and not including the 'stop #'
population = np.arange(1, 101)

#sample size
m = 10
sample = np.random.choice(a = population, size = m)
print("Sample :", sample)
```

Run

Reset

Select the answer matching your sample below.

○ -0.2144699617662135 0.4160333636063626 0.02927226924712613
-0.5072293848619751 2.6014747539872567 0.17141327084834654
-0.21195901381927462 -0.37671989689029883 0.1799644167541328
-0.8515596897956541

○ 110 67 93 99 103 18 84 107 58 87

○ 12 14 57 79 70 72 36 25 67 9

● 67 93 99 18 84 58 87 98 97 48

○ 0.70579387 -0.69160146 1.12461493 0.36499493 0.19864388 -0.85155969
 -2.88011494 -0.77227959 0.36499493 0.809468

○ 9 25 68 88 80 49 11 95 53 99

✓ **Correct**