3. Exercise Sheet **Summer Term 2023**
Statistical Methods for Data Analyses A **Prof. W. Rhode**
Submission: 09.05.2020 23:59 **Dr. M. Linhoff**

Time	Group	Submission in Moodle; Mails with subject: [SMD2023]
Th. 12:00–13:00	A	lukas.beiske@udo.edu and tristan.gradetzke@udo.edu
Fr. 08:45–09:45	B	jonas.hackfeld@ruhr-uni-bochum.de and ludwig.neste@udo.edu
Fr. 10:00–11:00	C	stefan.froese@udo.edu and vincent.latko@udo.edu

Exercise 5 *Random number generator* **5 p.**

Linear congruent random number generators generate a new integer random number from the previous one by the rule

$$x_n = (a \cdot x_{n-1} + c) \bmod m.$$

Division by m then yields a uniformly distributed real random number between 0 and 1.

In the following tasks the methods in the file

```
project_<groupname>/random.py
```

of your project will be added. To test your implementation you can use the script `exercises/lcg.py` with the command

```
$ # (replace <groupname> with the name of your group)
$ python exercises/lcg.py project_<groupname> -o lcg.pdf
```

If edited correctly, all tests should pass successfully. Keep in mind, however, that successfully passed tests do not equate to everything being correct!

(a) Implement the `advance` method of the class `LCG`.

(b) Determine the period length for different values for a by using the class you implemented with $c = 3$ and $m = 1024$.

What is the maximum period length?

For which values of a is the period length the longest?

(c) Implement the `uniform` method of class `LCG`, so that it generates uniformly distributed continuous numbers in the interval [`low`, `high`].

Include the overview PDF you created in your submission. Interpret the results.

For the following tasks, use a linear-congruent random number generator with the parameters $a = 1601$, $c = 3456$ and $m = 10\,000$. You can import your random number generator implemented in the previous task with the following command:

```
# (replace <groupname> with the name of your group)
from project_<groupname>.random import LCG
```

(d) Generate 10 000 uniformly distributed random numbers in the interval $[0, 1]$.

Plot these numbers in a suitable (100 bins) histogram.

Does the result meet the requirements of a good random number generator?

Does the behavior of the random generator depend on the initial value x_0, and if so, how?

(e) Plot pairs and triplets of consecutive random numbers as a two-dimensional and three-dimensional scatter plot respectively. Does the result meet the requirements of a good random number generator?

(f) Generate the same plot from **(d)** and **(e)** again by using the generator `np.random.default_rng`.

Listing 1: Example for a three dimensional scatterplot in `matplotlib`:

```
import matplotlib.pyplot as plt
import numpy as np

x, y, z = np.random.normal(size=(3, 1000))

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1, projection='3d')

ax.scatter(
    x, y, z,
    s=5,          # smaller points
    alpha=0.3,    # 70% transparency
)
# set the orientation of the 3d axis
ax.view_init(elev=30, azim=20)
plt.show()
```

Exercise 6 *Uniform distributions* **5 p.**

Given is a random number generator that generates uniformly distributed random numbers $u \in [0, 1]$.

Specify **efficient algorithms** with which you can generate random numbers that follow the specified distributions.

N is the normalization constant and has to be calculated.

Implement the corresponding procedures as methods of the class `Generator` in your project in the file

project_<groupname>/random.py

You can test your `generator` by executing the command

```
$ # (replace <groupname> with your group name)
$ python exercises/distributions.py project_<groupname> -o distributions.
   pdf
```

If edited correctly, all tests should pass successfully. Keep in mind, however, that successfully passed tests do not equate to everything being correct!

(a) Exponential distribution (method `exponential`)

$$f(x) = \begin{cases} Ne^{-x/\tau} & 0 <= x < \infty \\ 0 & \text{else} \end{cases} \tag{1}$$

(b) Power distribution with negative index ($n \geq 2$) (method `power`)

$$f(x) = \begin{cases} Nx^{-n} & x_{\text{min}} \leq x \leq x_{\text{max}} \\ 0 & \text{else} \end{cases} \tag{2}$$

(c) Cauchy-distribution (method `cauchy`)

$$f(x) = \frac{1}{\pi} \frac{1}{1+x^2}$$

Include the overview PDF you created in your submission. Interpret the results.