Εργαστήριο Πιθανότητες Αναφορά

Μπουικλής Άγγελος - Π2018076

28 Μαρτίου 2023

**Όλες οι ασκήσεις εκτελέστηκαν στη γλώσσα προγραμματισμού Python.*

Εργαστήριο - 01

Για τις ανάγκες του πρώτου εργαστηρίου δημιουργήθηκαν οι παρακάτω ασκήσεις σε μορφή συναρτήσεων στη γλώσσα Python.

Ένα πρόγραμμα το οποίο κάθε φορά που εκτελείται επιστρέφει:

1. *έναν τυχαίο αριθμό (από το 0 έως το sys.maxsize)*
2. *δέκα τυχαίους αριθμούς (από το 0 έως το sys.maxsize)*
3. *τον μέγιστο τυχαίο αριθμό*
4. *δέκα τυχαίους αριθμούς (από το 1 έως το 6) που στην ουσία προσομοιώνει τη ρίψη ενός ζαριού, και τα αποθηκεύει σε ένα αρχείο .CSV*

```
import random
import sys
import csv

#1
def rand():
    print("Random Number")
    print(random.randint(0, sys.maxsize))

#2
def rand10():
    print("10 Random Numbers")
    for i in range(10):
        print(random.randint(0, sys.maxsize))

#3
def randMAX():
    print("Random Max")\selectlanguage{english}
    print(sys.maxsize)
```

```
#4
def diceFreq():
    print("Random Dice Roll")

    freq = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0}
    for i in range(10):
            roll = (random.randint(1, 6))
            freq[roll] += 1
    with open(’dice_freq.csv’, ’w’, newline=’’) as file:
            writer = csv.writer(file)
            writer.writerow([’Roll Value’, ’Frequency’])
            for roll, count in freq.items():
                writer.writerow([roll, count])

if __name__ == "__main__":
    print(rand(), rand10(), randMAX(), diceFreq())
```

Παρακάτω το αρχείο .CSV όπου εκτελέστηκε ρίψη κέρματος 1.000.000 φορές και αρχίζουμε σιγά-σιγά να πλησιάζουμε στο σημείο όπου όλες οι πλευρές του ζαριού είναι ισοπίθανο να εμφανιστούν

	A	B
1	Roll Value	Frequency
2	1	16666569
3	2	16667027
4	3	16662492
5	4	16664917
6	5	16673797
7	6	16665198

Εργαστήριο - 02

Για τις ανάγκες του δεύτερου εργαστηρίου δημιουργήθηκε η παρακάτω ασκήσεις σε μορφή συναρτήσεων στη γλώσσα Python.

Ένα πρόγραμμα πρόγραμμα στο οποίο:

1. *εισάγει ο χρήστης πόσες φορές θα ρίξει ένα δίκαιο κέρμα*

2. *δήμιουργεί ένα αρχείο .CSVστο οποίο αποθηκεύει τα δεδομένα από την εκτέλεση του προγράμματος. Στο column 'Heads' αποθηκεύεται πόσες φορές ήρθε Κορώνα ενώ στο column 'Tails' αποθηκεύεται πόσες φορές ήρθε Γράμματα*

Στο value εισάγει ο χρήστης πόσες ρίψεις κέρματος επιθυμει να εκτελέσει, στο freq ορίζουμε την αρχιτεκτονική του αρχείου, στο coin δίνουμε τις δύο επιλογές από τις οποίες επιλέγει ισοπίθανα η random, στο φορ-λοοπ βάζουμε ως range το value έπειτα δημιουργούμε και εισάγουμε τα δεδομένα στο αρχείο .CSV

```
import random
import csv

print("Coin Flip")
value = input("How many times should i flip the Coin?\n")
value = int(value)
freq = {'Heads': 0, 'Tails': 0}
coin = ['Heads', 'Tails']
for i in range(value):
    roll = (random.choice(coin))
    freq[roll] += 1
with open('coin_Flip.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['Flip_Value', 'Frequency'])
    for roll, count in freq.items():
        writer.writerow([roll, count]
```

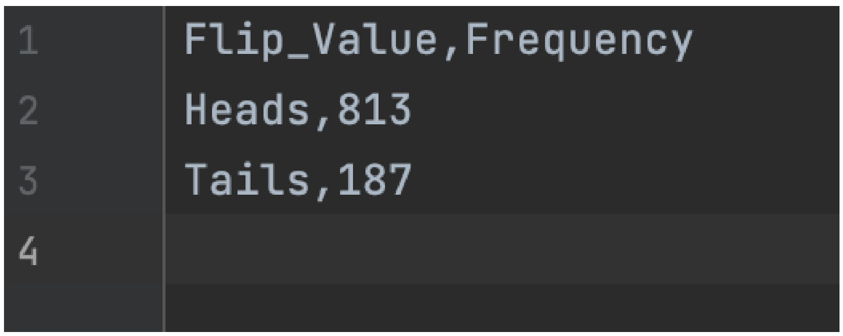
Όπως παρατηρήθηκε, διότι εκτέλεσα το πρόγραμμα αντίστοιχα για 10, 100, 1000, 10.000, 100.000 και 1.000.000 φορές, όσες περισσότερες ρίψεις πραγματοποιήσουμε, τόσο πιο πολύ πλησιάζουμε στην ιδανική πιθανότητα 50 Κορώνα και 50 Γράμματα, παρακάτω ο κώδικας:

Στο ίδιο εργαστήριο θα έπρεπε να τροποποιήσουμε τον παραπάνω κώδικα έτσι ώστε οι πιθανότητες αντί για 50-50 να είναι 80-20 αντίστοιχα για Κορώνα-Γράμματα

```
import random
import csv

print("Coin Flip")
value = input("How many times should i flip the Coin?\n")
value = int(value)
freq = {'Heads': 0, 'Tails': 0}
coin = ['Heads', 'Tails']
def coin_flip():
    if random.random() < 0.8:
        return 'Heads'
    else:
        return 'Tails'
```

```
for i in range(value):
    roll = coin_flip()
    freq[roll] += 1
with open('coin_Flip.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['Flip_Value', 'Frequency'])
    for roll, count in freq.items():
        writer.writerow([roll, count]
```

Η διαφορά φυσικά είναι εμφανής, στο παρακάτω στιγμιότυπο, όπου η Κορώνα ήρθε 813 φορές και τα Γράμματα 187



```
Flip_Value,Frequency
Heads,813
Tails,187
```

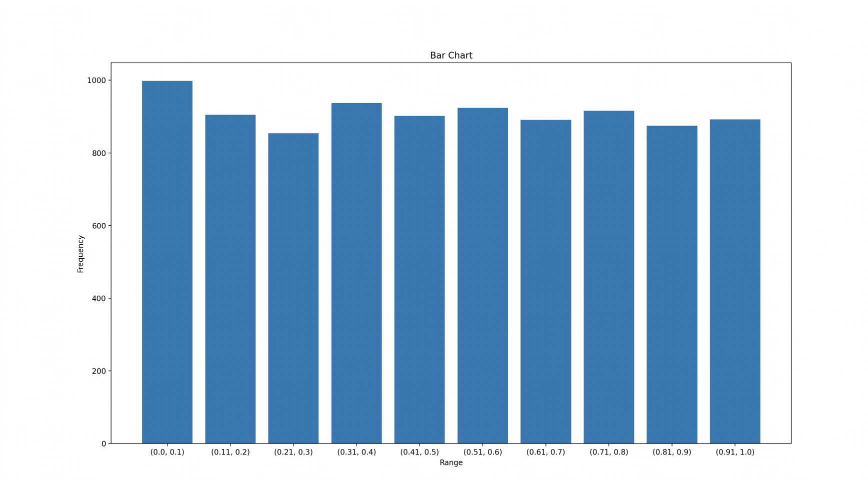
Εργαστήριο - 03

Για τις ανάγκες του τρίτου εργαστηρίου δημιουργήθηκε ο παρακάτω κώδικας στον οποίο ο χρήστης εισάγει πόσους τυχαίους αριθμούς επιθυμεί να τυπώσει το πρόγραμμα στο .CSV αρχείο.

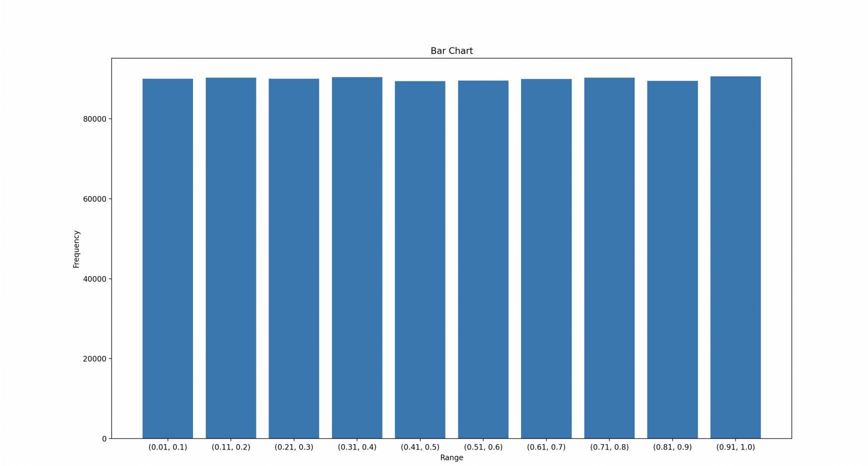
```
iimport random
import csv

value = input('How many random numbers do you need?\n')
value = int(value)
freq = {(0.01, 0.1): 0, (0.11, 0.2): 0, (0.21, 0.3): 0,
   (0.31, 0.4): 0, (0.41, 0.5): 0, (0.51, 0.6): 0, (0.61,
   0.7): 0, (0.71, 0.8): 0, (0.81, 0.9): 0, (0.91, 1.0): 0}
for i in range(value):
    roll = random.uniform(0, 1)
    for range in freq.keys():
        if range[0] <= roll <= range[1]:
            freq[range] += 1
            break
with open('RandomDouble.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['Range', 'Frequency'])
    for roll, count in freq.items():
```

```
        writer.writerow([roll, count])
```

Στις 1.000 φορές φαίνεται πως οι μπάρες είναι σχετικά ανοιμοιόμορφες:



Στις 1.000.000 φορές είναι εμφανής η διαφορά μιας και τείνουμε προς την ισοπίθανη κατανομή μιας και οι μπάρες δημιουργούν σχεδόν μια νοητή ευθεία στις κορυφές τους

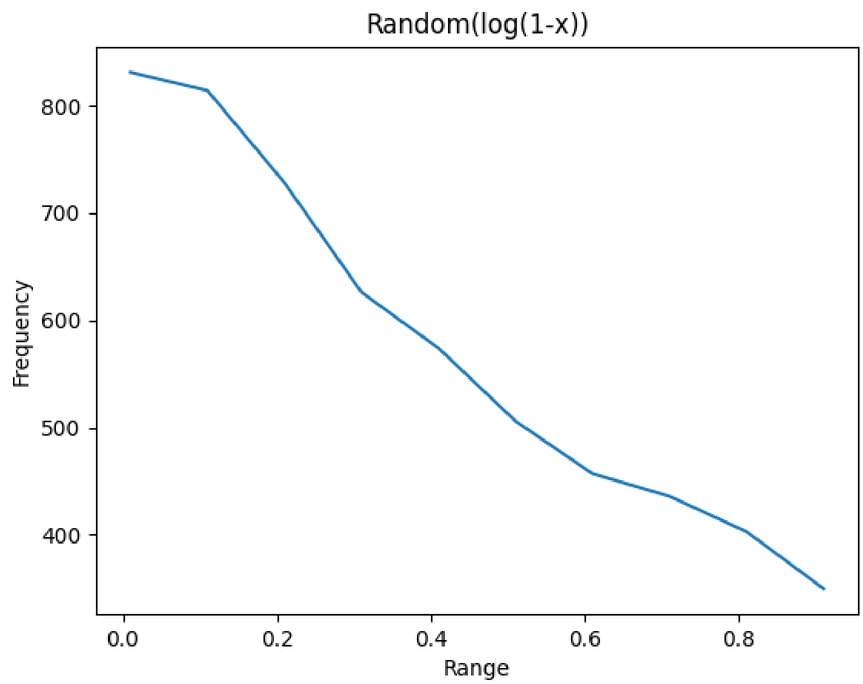


Συμπέρασμα: όπως παρατηρήθηκε όσο περισσότερες φορές επαναλαμβάνεται το πείραμα, τόσο περισσότερο τείνουμε προς σε μια ομοιόμορφη κατανομή. Για παράδειγμα στις 1.000.000 η νοητή ευθεία φαίνεται ξεκάθαρα πως είναι πιο ομοιόμορφη σε σχέση με αυτή των 1.000 επαναλήψεων.

Ερώτημα Β

i) y = -log(1-x)

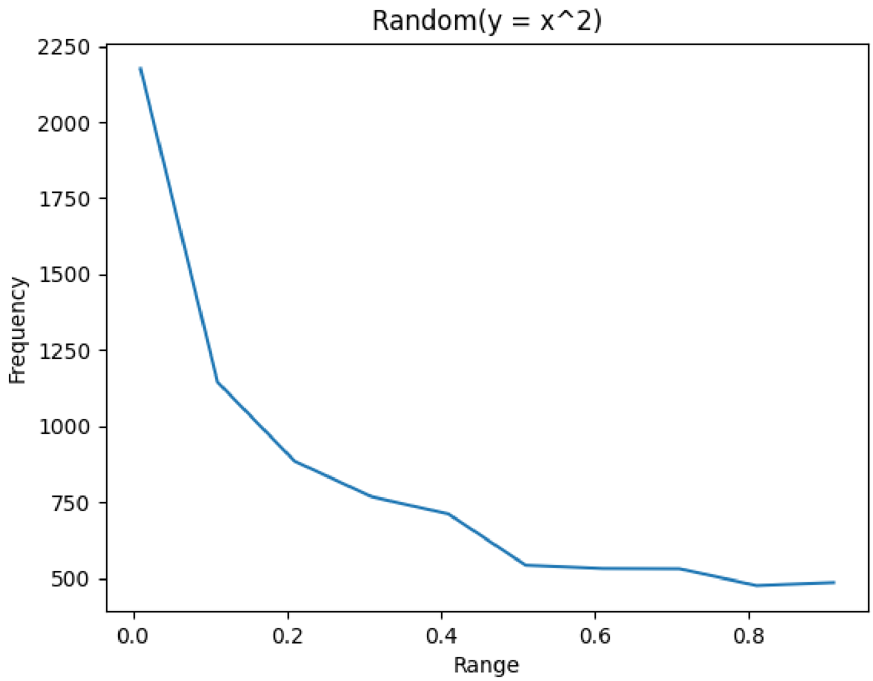
```
for i in range(10000):
    y = - math.log(1 - random.uniform(0, 1))
    for range in freq.keys():
        if range[0] <= y <= range[1]:
            freq[range] += 1
```



Συμπέρασμα: Όπως φαίνεται και στο ιστόγραμμα από τις 10.000 φορές που εκτελέστηκε το for-loop είναι ξεκάθαρο πως στο διάστημα (0.0 - 0.1) είναι οι περισσότεροι τυχαίοι αριθμοί για τους οποίους ισχύει y = -log(1-x)

ii) y $= x^2$

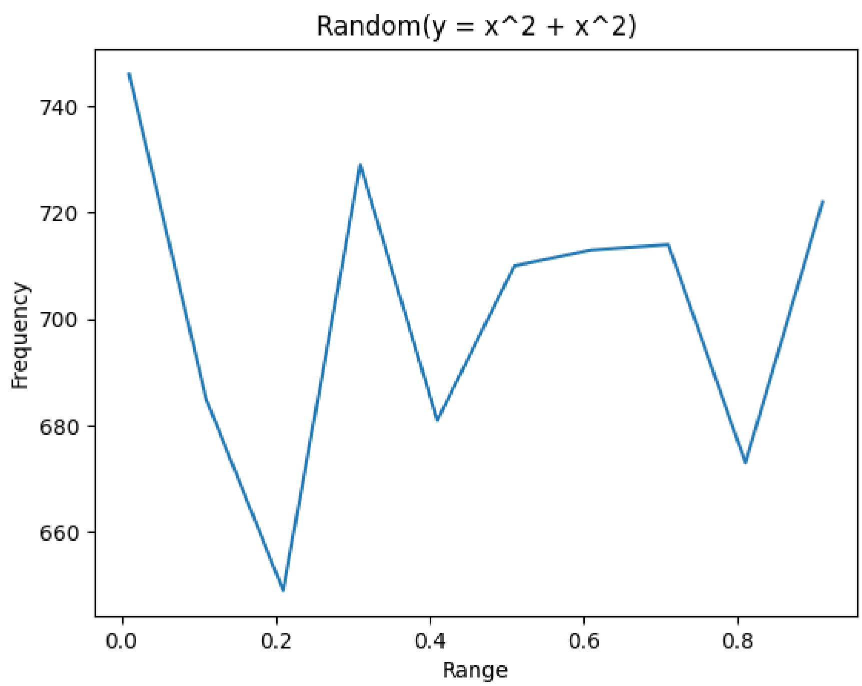
```
for i in range(10000):
    y = math.pow(random.uniform(0, 1), 2)
    for range in freq.keys():
        if range[0] <= y <= range[1]:
            freq[range] += 1
```



Συμπέρασμα: Όπως φαίνεται και στο ιστόγραμμα η κατανομή ταιριάζει με αυτή του x^2

iii) y $= (x^j)^2 + (x^k)^2$

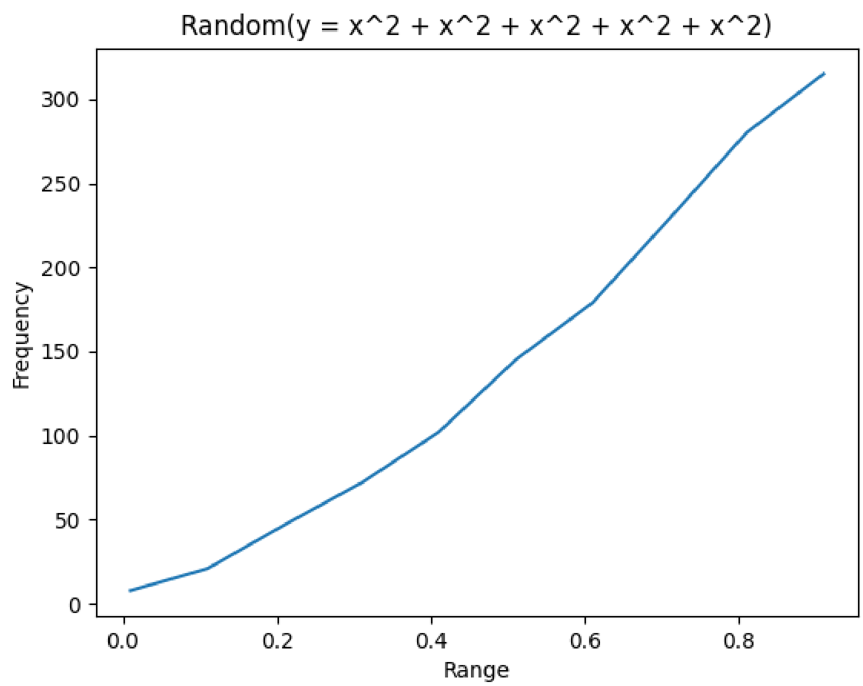
```
for i in range(10000):
    y = math.pow(random.uniform(0, 1), 2) + math.pow(random.
        uniform(0, 1), 2)
    for range in freq.keys():
        if range[0] <= y <= range[1]:
            freq[range] += 1
```



Συμπέρασμα: Όπως φαίνεται και στο ιστόγραμμα η κατανομή ταιριάζει με αυτή του x^2

iv) y $= (x^j)^2 + (x^k)^2 + (x^l)^2 + (x^m)^2 + (x^n)^2$

```
for i in range(10000):
    y = math.pow(random.uniform(0, 1), 2) + math.pow(random.
        uniform(0, 1), 2) + math.pow(random.uniform(0, 1), 2)
         + math.pow(random.uniform(0, 1), 2) + math.pow(
        random.uniform(0, 1), 2)
    for range in freq.keys():
        if range[0] <= y <= range[1]:
            freq[range] += 1
```



Εγαστήριο - 04

Για τις ανάγκες του εργαστηρίου αναπτύχθηκε κώδικας στον οποίο επιλέγει ο χρήστης το πλήθος των αριθμών εκ των οποίων θα υπολογίσουμε την μέση τιμή, την διασπορά και την τυπική απόκλιση. Βάζοντας τους αρχικά σε ένα πίνακα κι έπειτα κάνοντας του υπολογισμούς.

```python
import random
import math

value = int(input("How many random numbers do you want to
    generate? "))

random_numbers = [random.randint(1, 100) for i in range(
    value)]

mean = sum(random_numbers) / value

variance = sum([((x - mean) ** 2) for x in random_numbers])
    / value

std_dev = math.sqrt(variance)

print("Random numbers: ", random_numbers)
print("Mean: ", mean)
print("Variance: ", variance)
print("Standard deviation: ", std_dev)
```

Στο ιστόγραμμα φαίνεται η κόκκινη γραμμή η οποία αναπαριστά τη μέση τιμή, οι μπλε κουκίδες ως τυχαίες τιμές και το πλαίσιο ως η τυπική απόκλιση