Generator liczb losowych

Michał Kuliński

1 Wstep

W projekcie zaimplementowany został generator liczb pseudolosowych MT19937 (Mersenne Twister), który oparty jest na arytmetyce modularnej. Na bazie tego generatora stworzone zostały rozkłady: jednostajny, Bernoulliego, dwumianowy, Poissona, wykładniczy oraz normalny.

2 Hipotezy badawcze

- Implementacja generatora bez wykorzystania dostepnych funkcji czy bibliotek dla generatorów liczb losowych
- Implementacja generatora bez wykorzystania zegara systemowego
- Przechodzenie przez generator testów Dieharder

3 MT19937

3.1 Opis

Generator liczb pseudolosowych, który na długość okresu wybiera jedna z liczb pierwszych Mersenne'a. Istnieja dwa warianty algorytmu, starszy oraz mniej używany z 64-bitowym słowem (MT19937-64) oraz nowszy (zaimplementowany w tym projekcie) oznaczany poprzez MT19937 z 32-bitowa długościa słowa.

3.2 Działanie oraz kod

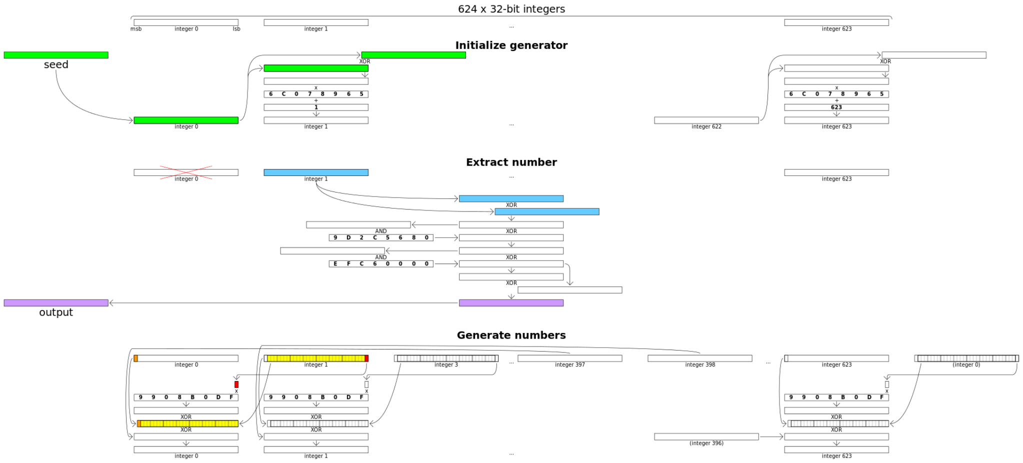


Figure 1: Działanie MT19937 (via wikipedia.org)

```
# MersenneTwister/MT19937.py
class MT19937:
    """Mersenne Twister standard implementation (MT19937)"""

    def __init__(self, seed):
        """Initialize standard coefficients for MT19937"""

        # w: word size (in number of bits)
        self.w = 32

        # n: degree of recurrence
        self.n = 624

        # m: middle word, an offset used in the recurrence relation defining the series x,
        self.m = 397

        # r: separation point of one word, or the number of bits of the lower bitmask, 0 <=
        self.r = 31

        # a: coefficients of the rational normal form twist matrix
        self.a = 2567483615  # 0x9908B0DF

        # u,d,l: additional Mersenne Twister tempering bit shifts / masks
        self.u = 11
        self.d = 4294967295  # 0xFFFFFFFF
        self.l = 18

        # s,t: tempering bit shifts
        self.s = 7
        self.t = 15

        # b,c: tempering bitmasks
        self.b = 2636928640  # 0x9D2C5680
        self.c = 4022730752  # 0xEFC60000

        # f: generator parameter
        self.f = 1812433253

        # Auxiliary constants
        self.RAND_MAX = 2 ** self.w - 1
        self.RAND_MIN = 0

        # -- Miscellaneous variables -- #

        # Array of length n to store the state of the generator
        self.MT = []
        self.index = self.n + 1

        self.lower_mask = (1 << self.r) - 1
        # lowest w bits - ((1 << w) - 1) is a mask to isolate w bits
        self.upper_mask = (~self.lower_mask) & ((1 << self.w) - 1)

        # -- Initialize generator from a seed -- #
        self.seed = seed
        self.index = self.n
        self.MT.append(seed)
        for i in range(1, self.n):
            mask = (1 << self.w) - 1  # mask to isolate w bits
            self.MT.append((self.f * (self.MT[i - 1] ^ (self.MT[i - 1] >> (self.w - 2))) + i

    def extractNumber(self):
        """Extract a tempered value based on MT[index]"""
        if self.index >= self.n:
            if self.index > self.n:
                raise RuntimeError("Generator was never seeded")
            self.generateNumbers()

        y = self.MT[self.index]
        y ^= ((y >> self.u) & self.d)
        y ^= ((y << self.s) & self.b)
        y ^= ((y << self.t) & self.c)
        y ^= (y >> self.l)

        self.index += 1
        return y & ((1 << self.w) - 1)

    def generateNumbers(self):
        """Generate the next n values from the series x_i"""
        for i in range(0, self.n):
            x = (self.MT[i] & self.upper_mask) + (self.MT[(i + 1) % self.n] & self.lower_mas
            xa = x >> 1
            if x % 2 != 0:
                xa = xa ^ self.a
            self.MT[i] = self.MT[(i + self.m) % self.n] ^ xa

        self.index = 0

    def randomUnif(self):
        return self.extractNumber() / 2 ** self.w
```

4 Eksperymenty

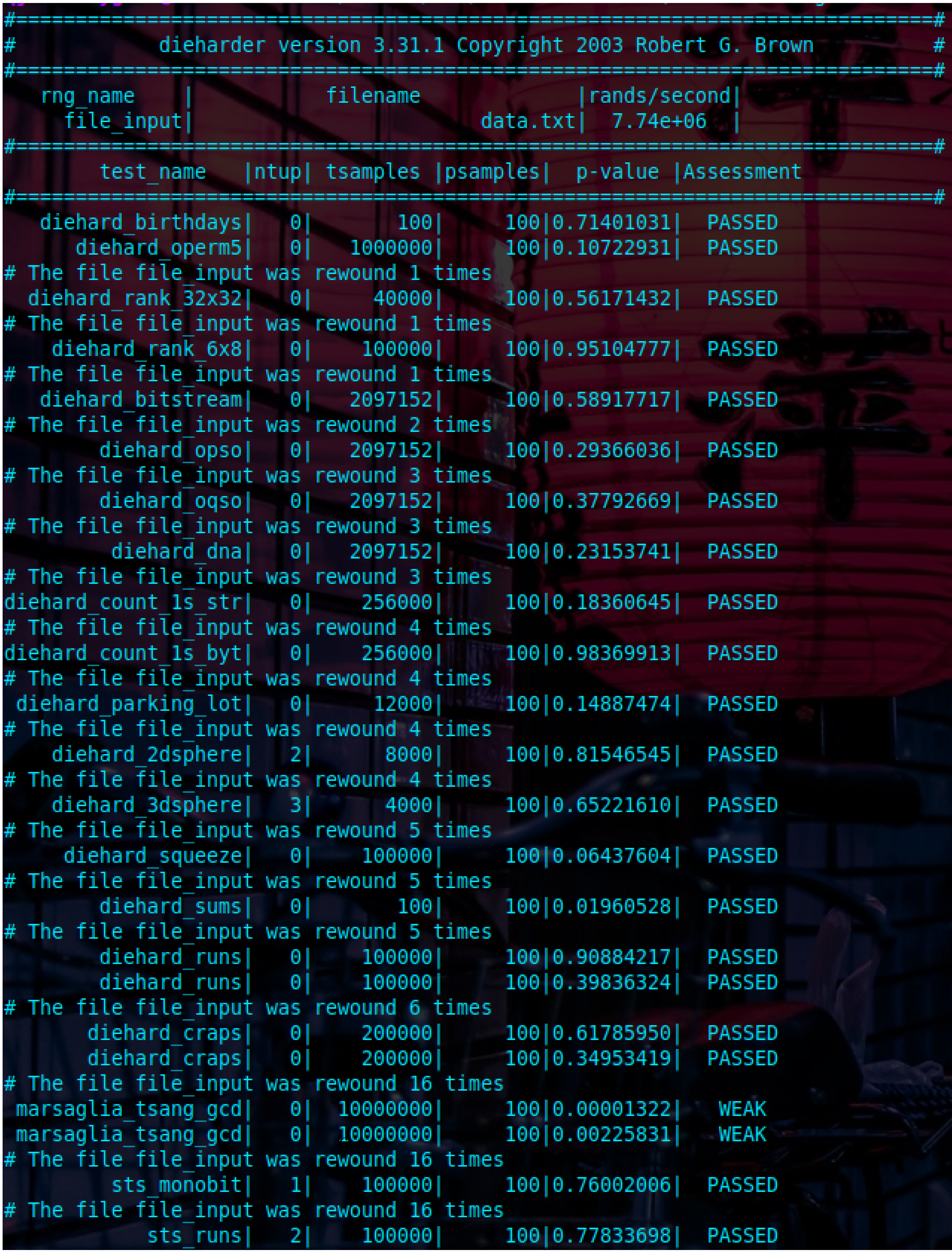
Eksperyment opiera sie na użyciu Diehardera na dużej próbce liczb wylosowanych z generatora. W zwiazku z tym program posiada funkcje która generuje podana liczbe liczb a nastepnie zapisuje je w pliku "data.txt". Ponadto funkcja ta używa specjalnego formatu pliku, który wymagany jest przy używaniu Diehardera. Wszystkie liczby sa 32 bitowymi intami, które tworzone sa przy użyciu biblioteki numpy - generowane liczby sa "konstruowane" przy użyciu numpy.int32 a nastepnie zapisywane do pliku (po obowiazkowym nagłówku). Tak wygenerowany plik jest gotowy do sprawdzenia przez Diehardera.

4.1 Kod generujacy plik "data.txt"

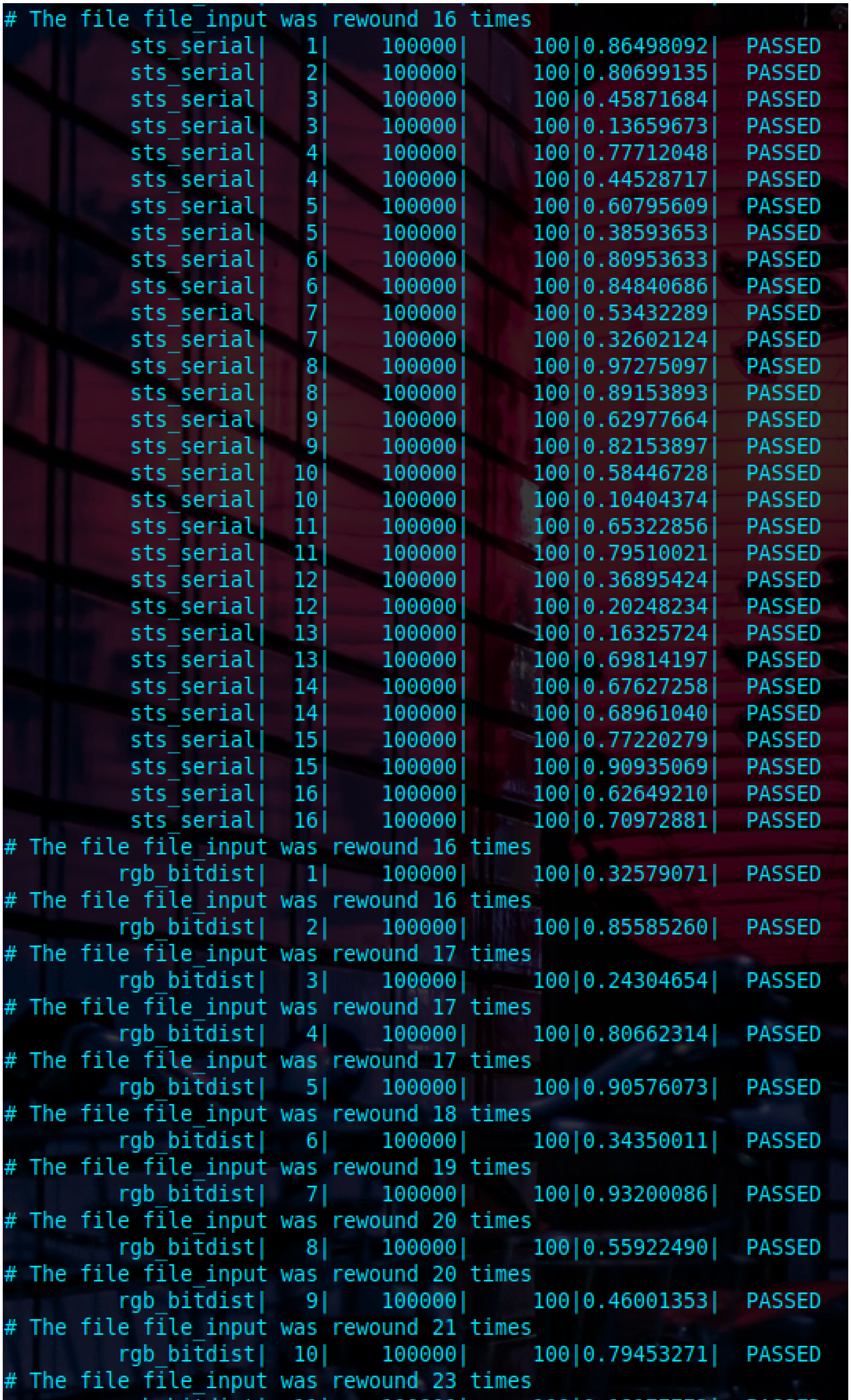
```
# main.py
def generateFileWithRandomNums(gen, amount):
    file = open("data.txt", "w")
    file.write("#==================================================================\n")
    file.write("# Generator MT19937 seed=" + str(gen.seed) + "\n")
    file.write("#==================================================================\n")
    file.write("type: d\n")
    file.write("count: " + str(amount) + "\n")
    file.write("numbit: 32\n")
    for i in range(amount):
        file.write(str(np.uint32(gen.extractNumber())) + "\n")
```

4.2 Wyniki testów Dieharder

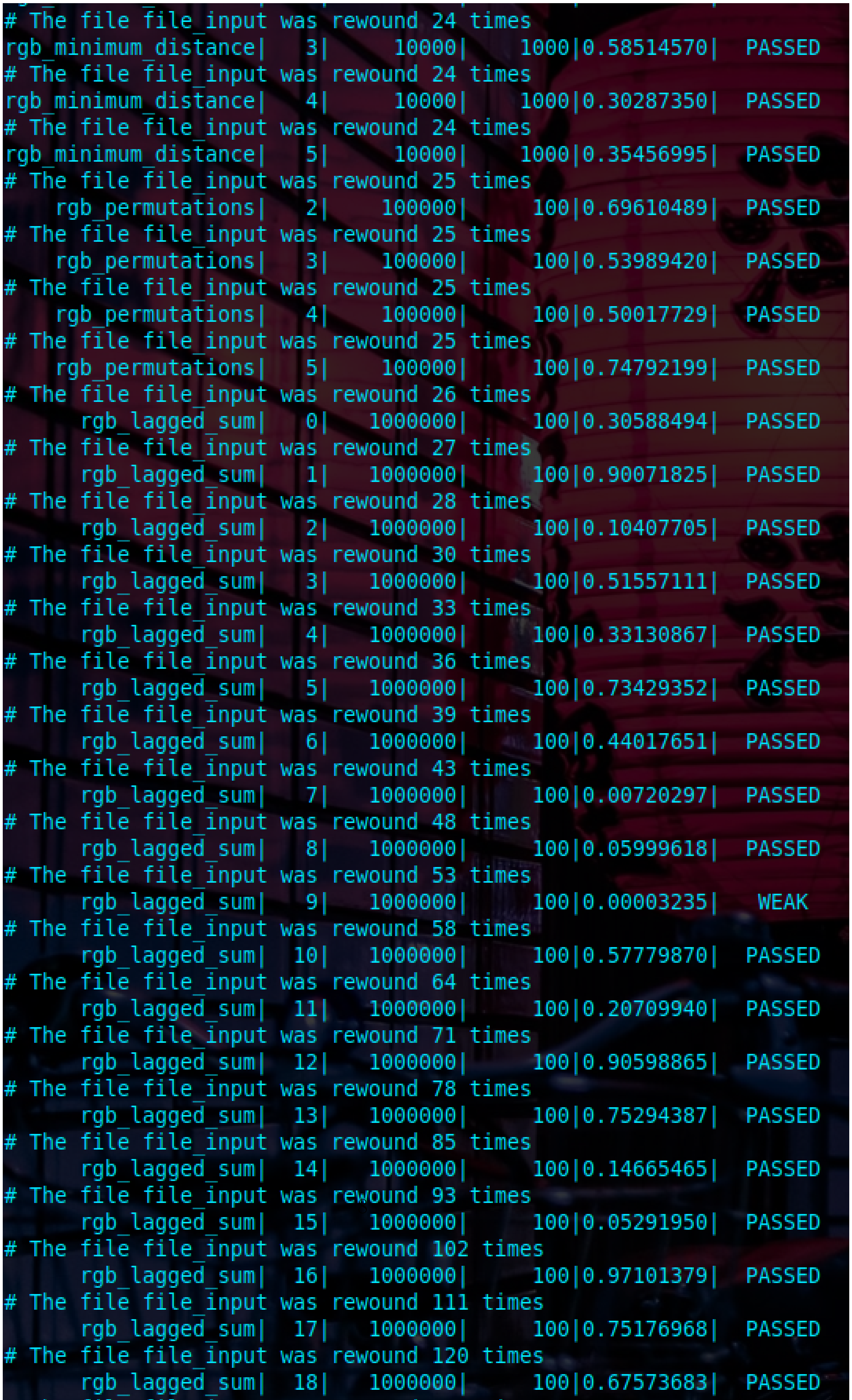
Poniższe wyniki testów sa wynikiem użycie Diehardera na pliku w którym wygenerowane zostało 200 milionów liczb. Wygenerowane liczby przeszły prawie wszystkie testy, pozostawiajac niektóre z oznaczeniem "WEAK". Testy te jednak dawały wynik "WEAK" tylko z powodu za małej liczby próbek. Zauważmy, że plik był w tych testach wielokrotnie "zapetlany", przez co liczby "przestawały być" losowe. Ponadto żaden z testów nie został oznaczony jako "FAILED".



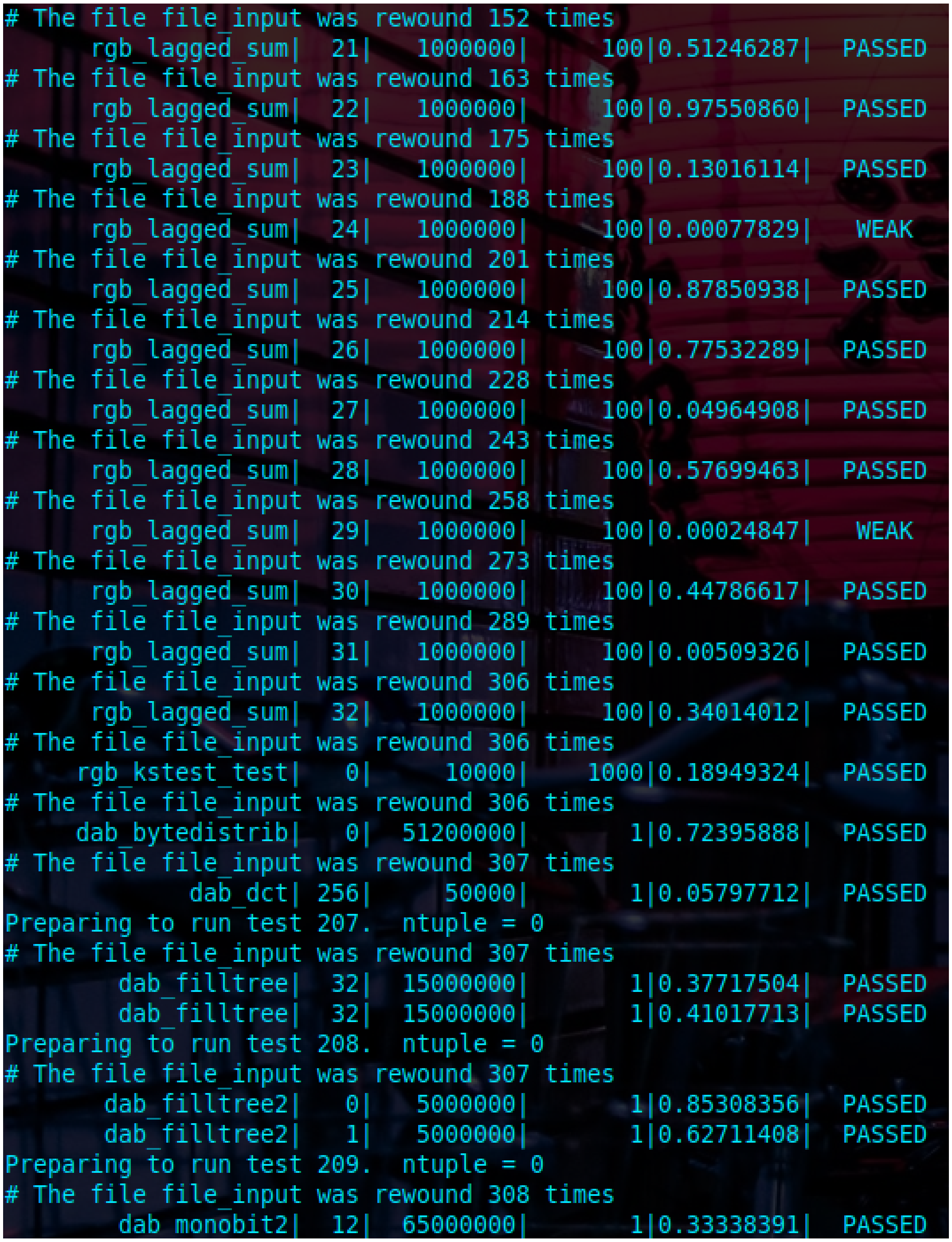
```
#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
   rng_name    |           filename             |rands/second|
     file_input|                        data.txt|  7.74e+06  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
   diehard_birthdays|   0|       100|     100|0.71401031|  PASSED
      diehard_operm5|   0|   1000000|     100|0.10722931|  PASSED
# The file file_input was rewound 1 times
  diehard_rank_32x32|   0|     40000|     100|0.56171432|  PASSED
# The file file_input was rewound 1 times
    diehard_rank_6x8|   0|    100000|     100|0.95104777|  PASSED
# The file file_input was rewound 1 times
   diehard_bitstream|   0|   2097152|     100|0.58917717|  PASSED
# The file file_input was rewound 2 times
        diehard_opso|   0|   2097152|     100|0.29366036|  PASSED
# The file file_input was rewound 3 times
        diehard_oqso|   0|   2097152|     100|0.37792669|  PASSED
# The file file_input was rewound 3 times
         diehard_dna|   0|   2097152|     100|0.23153741|  PASSED
# The file file_input was rewound 3 times
diehard_count_1s_str|   0|    256000|     100|0.18360645|  PASSED
# The file file_input was rewound 4 times
diehard_count_1s_byt|   0|    256000|     100|0.98369913|  PASSED
# The file file_input was rewound 4 times
 diehard_parking_lot|   0|     12000|     100|0.14887474|  PASSED
# The file file_input was rewound 4 times
    diehard_2dsphere|   2|      8000|     100|0.81546545|  PASSED
# The file file_input was rewound 4 times
    diehard_3dsphere|   3|      4000|     100|0.65221610|  PASSED
# The file file_input was rewound 5 times
     diehard_squeeze|   0|    100000|     100|0.06437604|  PASSED
# The file file_input was rewound 5 times
        diehard_sums|   0|       100|     100|0.01960528|  PASSED
# The file file_input was rewound 5 times
        diehard_runs|   0|    100000|     100|0.90884217|  PASSED
        diehard_runs|   0|    100000|     100|0.39836324|  PASSED
# The file file_input was rewound 6 times
       diehard_craps|   0|    200000|     100|0.61785950|  PASSED
       diehard_craps|   0|    200000|     100|0.34953419|  PASSED
# The file file_input was rewound 16 times
 marsaglia_tsang_gcd|   0|  10000000|     100|0.00001322|   WEAK
 marsaglia_tsang_gcd|   0|  10000000|     100|0.00225831|   WEAK
# The file file_input was rewound 16 times
         sts_monobit|   1|    100000|     100|0.76002006|  PASSED
# The file file_input was rewound 16 times
            sts_runs|   2|    100000|     100|0.77833698|  PASSED
```



```
# The file file_input was rewound 16 times
          sts_serial|   1|    100000|     100|0.86498092|  PASSED
          sts_serial|   2|    100000|     100|0.80699135|  PASSED
          sts_serial|   3|    100000|     100|0.45871684|  PASSED
          sts_serial|   3|    100000|     100|0.13659673|  PASSED
          sts_serial|   4|    100000|     100|0.77712048|  PASSED
          sts_serial|   4|    100000|     100|0.44528717|  PASSED
          sts_serial|   5|    100000|     100|0.60795609|  PASSED
          sts_serial|   5|    100000|     100|0.38593653|  PASSED
          sts_serial|   6|    100000|     100|0.80953633|  PASSED
          sts_serial|   6|    100000|     100|0.84840686|  PASSED
          sts_serial|   7|    100000|     100|0.53432289|  PASSED
          sts_serial|   7|    100000|     100|0.32602124|  PASSED
          sts_serial|   8|    100000|     100|0.97275097|  PASSED
          sts_serial|   8|    100000|     100|0.89153893|  PASSED
          sts_serial|   9|    100000|     100|0.62977664|  PASSED
          sts_serial|   9|    100000|     100|0.82153897|  PASSED
          sts_serial|  10|    100000|     100|0.58446728|  PASSED
          sts_serial|  10|    100000|     100|0.10404374|  PASSED
          sts_serial|  11|    100000|     100|0.65322856|  PASSED
          sts_serial|  11|    100000|     100|0.79510021|  PASSED
          sts_serial|  12|    100000|     100|0.36895424|  PASSED
          sts_serial|  12|    100000|     100|0.20248234|  PASSED
          sts_serial|  13|    100000|     100|0.16325724|  PASSED
          sts_serial|  13|    100000|     100|0.69814197|  PASSED
          sts_serial|  14|    100000|     100|0.67627258|  PASSED
          sts_serial|  14|    100000|     100|0.68961040|  PASSED
          sts_serial|  15|    100000|     100|0.77220279|  PASSED
          sts_serial|  15|    100000|     100|0.90935069|  PASSED
          sts_serial|  16|    100000|     100|0.62649210|  PASSED
          sts_serial|  16|    100000|     100|0.70972881|  PASSED
# The file file_input was rewound 16 times
         rgb_bitdist|   1|    100000|     100|0.32579071|  PASSED
# The file file_input was rewound 16 times
         rgb_bitdist|   2|    100000|     100|0.85585260|  PASSED
# The file file_input was rewound 17 times
         rgb_bitdist|   3|    100000|     100|0.24304654|  PASSED
# The file file_input was rewound 17 times
         rgb_bitdist|   4|    100000|     100|0.80662314|  PASSED
# The file file_input was rewound 17 times
         rgb_bitdist|   5|    100000|     100|0.90576073|  PASSED
# The file file_input was rewound 18 times
         rgb_bitdist|   6|    100000|     100|0.34350011|  PASSED
# The file file_input was rewound 19 times
         rgb_bitdist|   7|    100000|     100|0.93200086|  PASSED
# The file file_input was rewound 20 times
         rgb_bitdist|   8|    100000|     100|0.55922490|  PASSED
# The file file_input was rewound 20 times
         rgb_bitdist|   9|    100000|     100|0.46001353|  PASSED
# The file file_input was rewound 21 times
         rgb_bitdist|  10|    100000|     100|0.79453271|  PASSED
# The file file_input was rewound 23 times
```



```
# The file file_input was rewound 24 times
rgb_minimum_distance|   3|     10000|    1000|0.58514570|  PASSED
# The file file_input was rewound 24 times
rgb_minimum_distance|   4|     10000|    1000|0.30287350|  PASSED
# The file file_input was rewound 24 times
rgb_minimum_distance|   5|     10000|    1000|0.35456995|  PASSED
# The file file_input was rewound 25 times
    rgb_permutations|   2|    100000|     100|0.69610489|  PASSED
# The file file_input was rewound 25 times
    rgb_permutations|   3|    100000|     100|0.53989420|  PASSED
# The file file_input was rewound 25 times
    rgb_permutations|   4|    100000|     100|0.50017729|  PASSED
# The file file_input was rewound 25 times
    rgb_permutations|   5|    100000|     100|0.74792199|  PASSED
# The file file_input was rewound 26 times
      rgb_lagged_sum|   0|   1000000|     100|0.30588494|  PASSED
# The file file_input was rewound 27 times
      rgb_lagged_sum|   1|   1000000|     100|0.90071825|  PASSED
# The file file_input was rewound 28 times
      rgb_lagged_sum|   2|   1000000|     100|0.10407705|  PASSED
# The file file_input was rewound 30 times
      rgb_lagged_sum|   3|   1000000|     100|0.51557111|  PASSED
# The file file_input was rewound 33 times
      rgb_lagged_sum|   4|   1000000|     100|0.33130867|  PASSED
# The file file_input was rewound 36 times
      rgb_lagged_sum|   5|   1000000|     100|0.73429352|  PASSED
# The file file_input was rewound 39 times
      rgb_lagged_sum|   6|   1000000|     100|0.44017651|  PASSED
# The file file_input was rewound 43 times
      rgb_lagged_sum|   7|   1000000|     100|0.00720297|  PASSED
# The file file_input was rewound 48 times
      rgb_lagged_sum|   8|   1000000|     100|0.05999618|  PASSED
# The file file_input was rewound 53 times
      rgb_lagged_sum|   9|   1000000|     100|0.00003235|   WEAK
# The file file_input was rewound 58 times
      rgb_lagged_sum|  10|   1000000|     100|0.57779870|  PASSED
# The file file_input was rewound 64 times
      rgb_lagged_sum|  11|   1000000|     100|0.20709940|  PASSED
# The file file_input was rewound 71 times
      rgb_lagged_sum|  12|   1000000|     100|0.90598865|  PASSED
# The file file_input was rewound 78 times
      rgb_lagged_sum|  13|   1000000|     100|0.75294387|  PASSED
# The file file_input was rewound 85 times
      rgb_lagged_sum|  14|   1000000|     100|0.14665465|  PASSED
# The file file_input was rewound 93 times
      rgb_lagged_sum|  15|   1000000|     100|0.05291950|  PASSED
# The file file_input was rewound 102 times
      rgb_lagged_sum|  16|   1000000|     100|0.97101379|  PASSED
# The file file_input was rewound 111 times
      rgb_lagged_sum|  17|   1000000|     100|0.75176968|  PASSED
# The file file_input was rewound 120 times
      rgb_lagged_sum|  18|   1000000|     100|0.67573683|  PASSED
```



```
# The file file_input was rewound 152 times
      rgb_lagged_sum|  21|   1000000|     100|0.51246287|  PASSED
# The file file_input was rewound 163 times
      rgb_lagged_sum|  22|   1000000|     100|0.97550860|  PASSED
# The file file_input was rewound 175 times
      rgb_lagged_sum|  23|   1000000|     100|0.13016114|  PASSED
# The file file_input was rewound 188 times
      rgb_lagged_sum|  24|   1000000|     100|0.00077829|   WEAK
# The file file_input was rewound 201 times
      rgb_lagged_sum|  25|   1000000|     100|0.87850938|  PASSED
# The file file_input was rewound 214 times
      rgb_lagged_sum|  26|   1000000|     100|0.77532289|  PASSED
# The file file_input was rewound 228 times
      rgb_lagged_sum|  27|   1000000|     100|0.04964908|  PASSED
# The file file_input was rewound 243 times
      rgb_lagged_sum|  28|   1000000|     100|0.57699463|  PASSED
# The file file_input was rewound 258 times
      rgb_lagged_sum|  29|   1000000|     100|0.00024847|   WEAK
# The file file_input was rewound 273 times
      rgb_lagged_sum|  30|   1000000|     100|0.44786617|  PASSED
# The file file_input was rewound 289 times
      rgb_lagged_sum|  31|   1000000|     100|0.00509326|  PASSED
# The file file_input was rewound 306 times
      rgb_lagged_sum|  32|   1000000|     100|0.34014012|  PASSED
# The file file_input was rewound 306 times
     rgb_kstest_test|   0|     10000|    1000|0.18949324|  PASSED
# The file file_input was rewound 306 times
     dab_bytedistrib|   0|  51200000|       1|0.72395888|  PASSED
# The file file_input was rewound 307 times
             dab_dct| 256|     50000|       1|0.05797712|  PASSED
Preparing to run test 207.  ntuple = 0
# The file file_input was rewound 307 times
        dab_filltree|  32|  15000000|       1|0.37717504|  PASSED
        dab_filltree|  32|  15000000|       1|0.41017713|  PASSED
Preparing to run test 208.  ntuple = 0
# The file file_input was rewound 307 times
       dab_filltree2|   0|   5000000|       1|0.85308356|  PASSED
       dab_filltree2|   1|   5000000|       1|0.62711408|  PASSED
Preparing to run test 209.  ntuple = 0
# The file file_input was rewound 308 times
        dab_monobit2|  12|  65000000|       1|0.33338391|  PASSED
```

5 Wykresy rozkładów

Każdy rozkład ma odpowiadajacy mu plik z funkcjami, które tworza wykres danego rozkładu. Wykresy sa zmienialne, co znaczy że dla tego samego rozkładu możemy tworzyć wiele wykresów o różnych parametrach.

5.1 Rozkład jednostajny

5.1.1 Wartości domyślne (low=0, high=1)

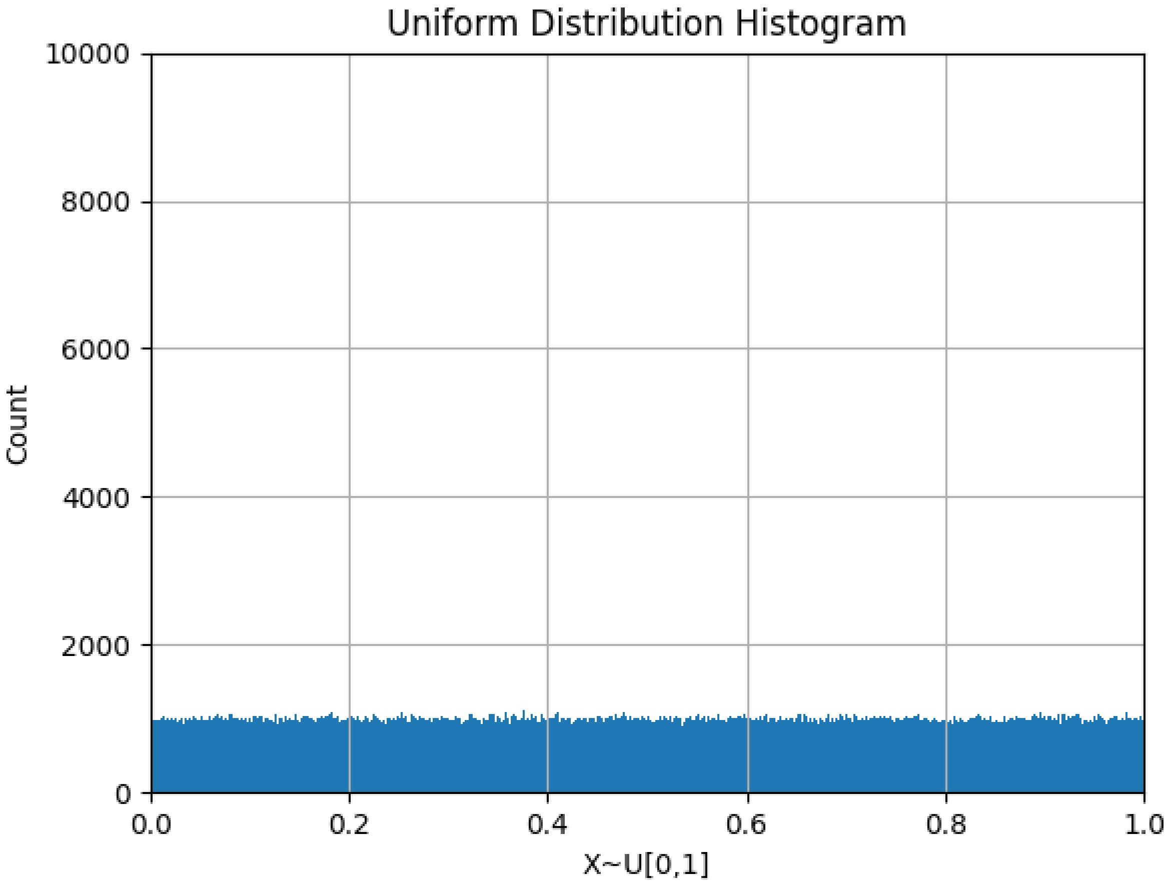


Figure 2: amount=1000000, bins=1000, yend=10000

5.1.2 Używanie niestandardowych wartości (low=1000, high=2000)

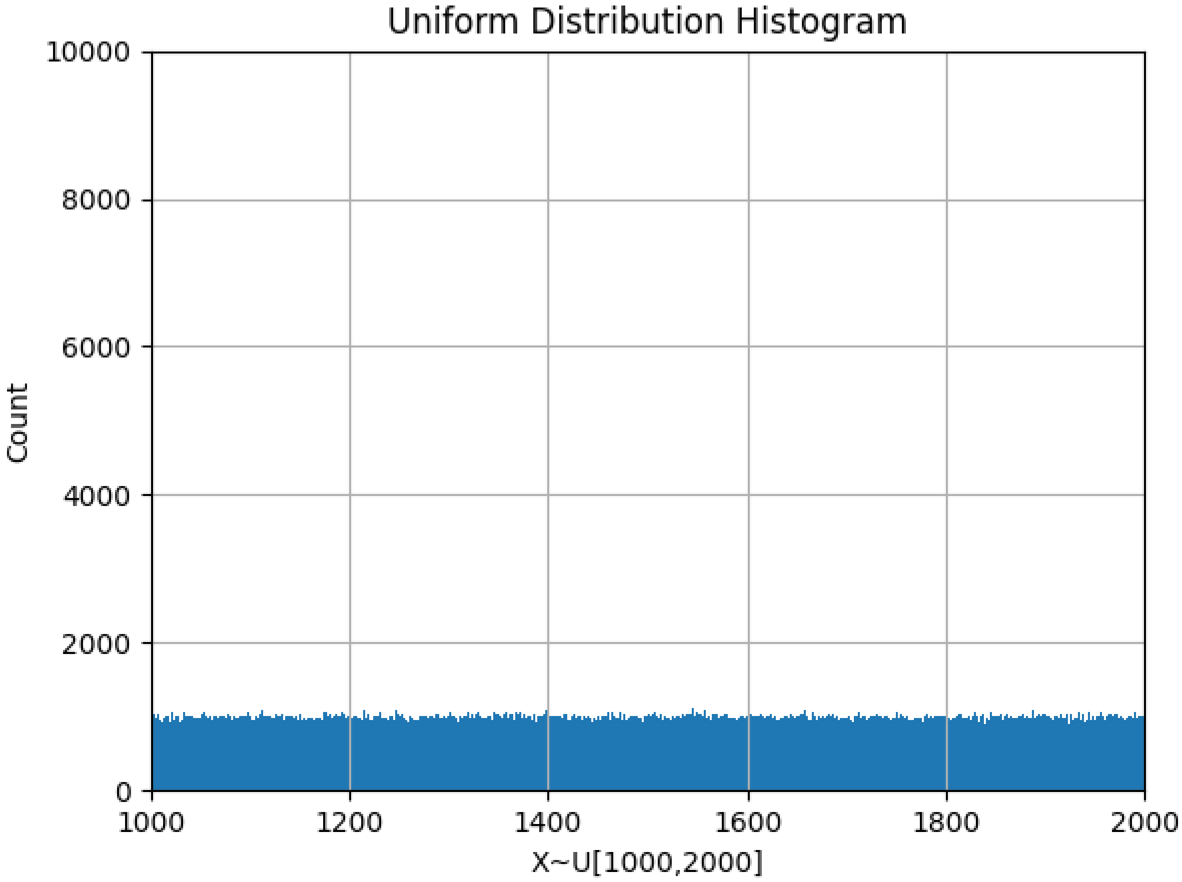


Figure 3: amount=1000000, bins=1000, yend=10000

5.2 Rozkład Bernoulliego

5.2.1 p = 0.5

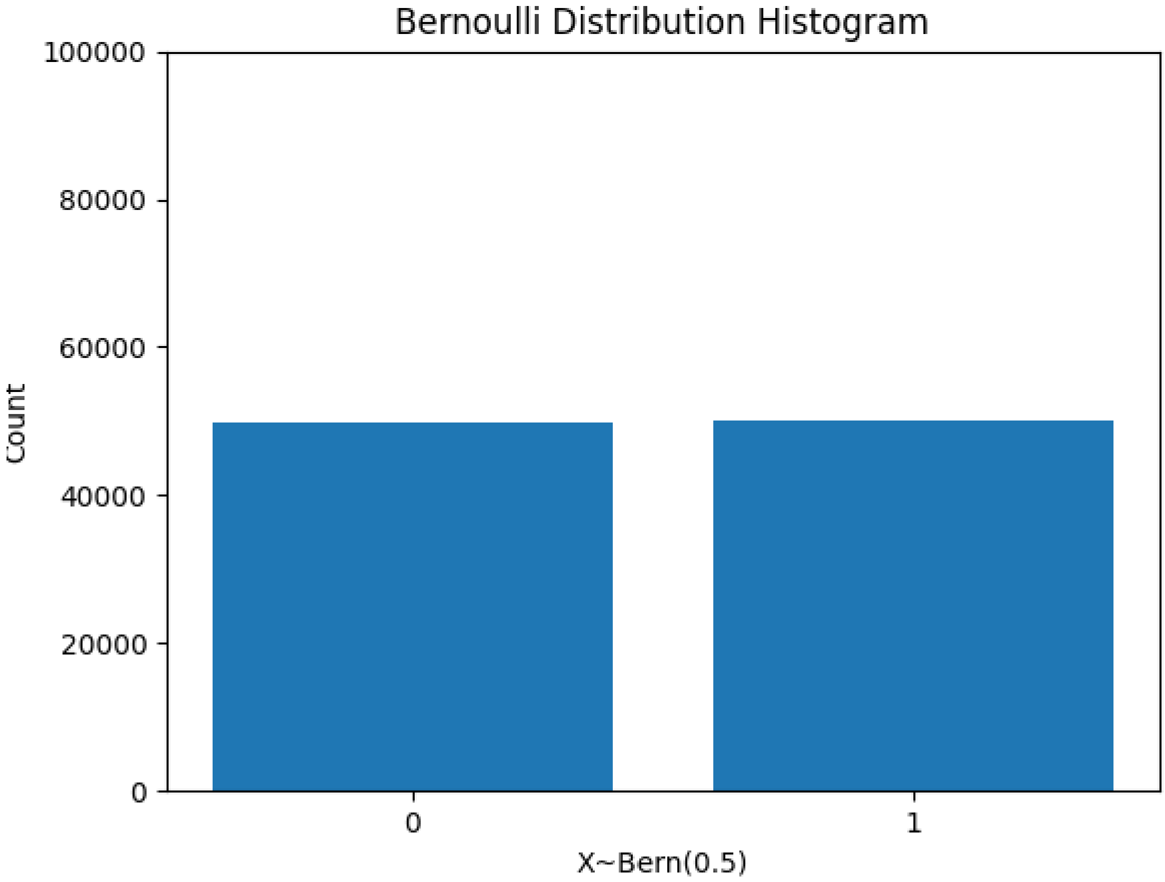


Figure 4: amount=100000

5.2.2 p = 0.1

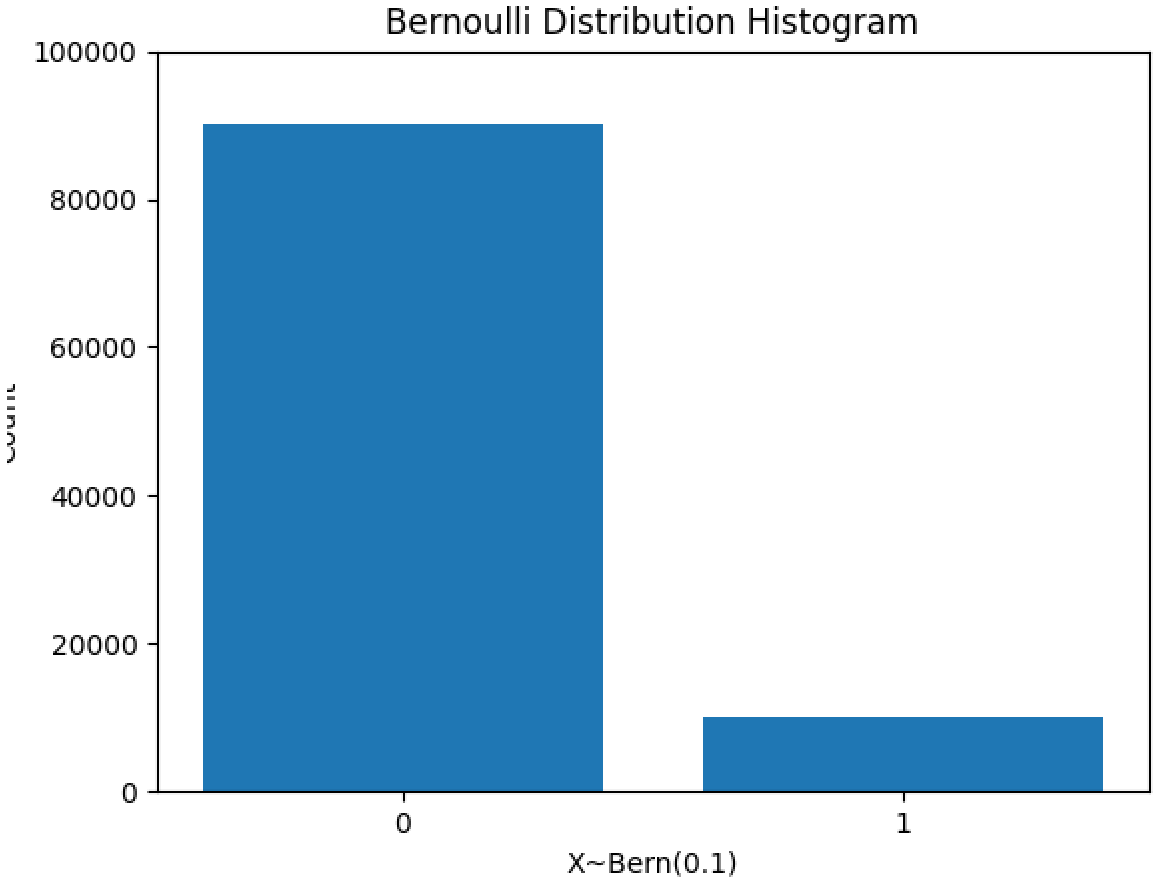


Figure 5: amount=100000

5.2.3 p = 0.9

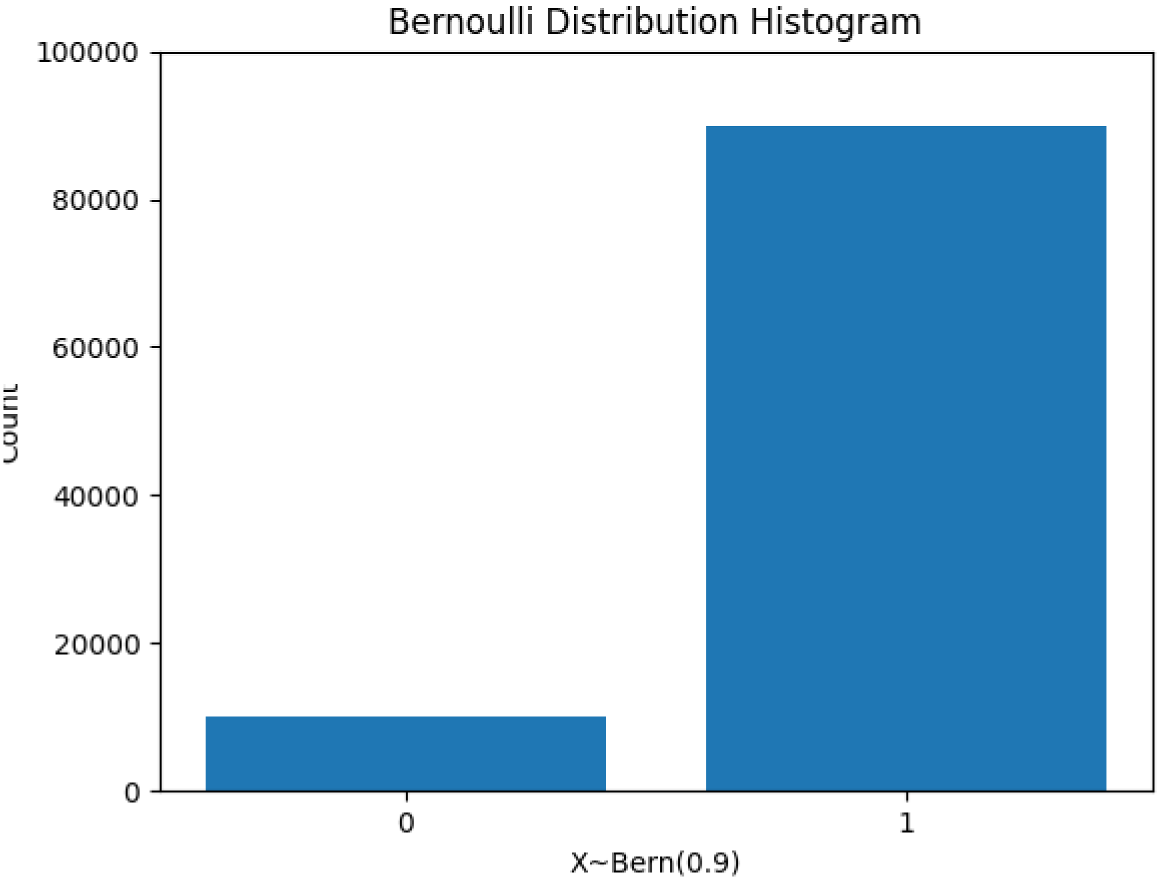
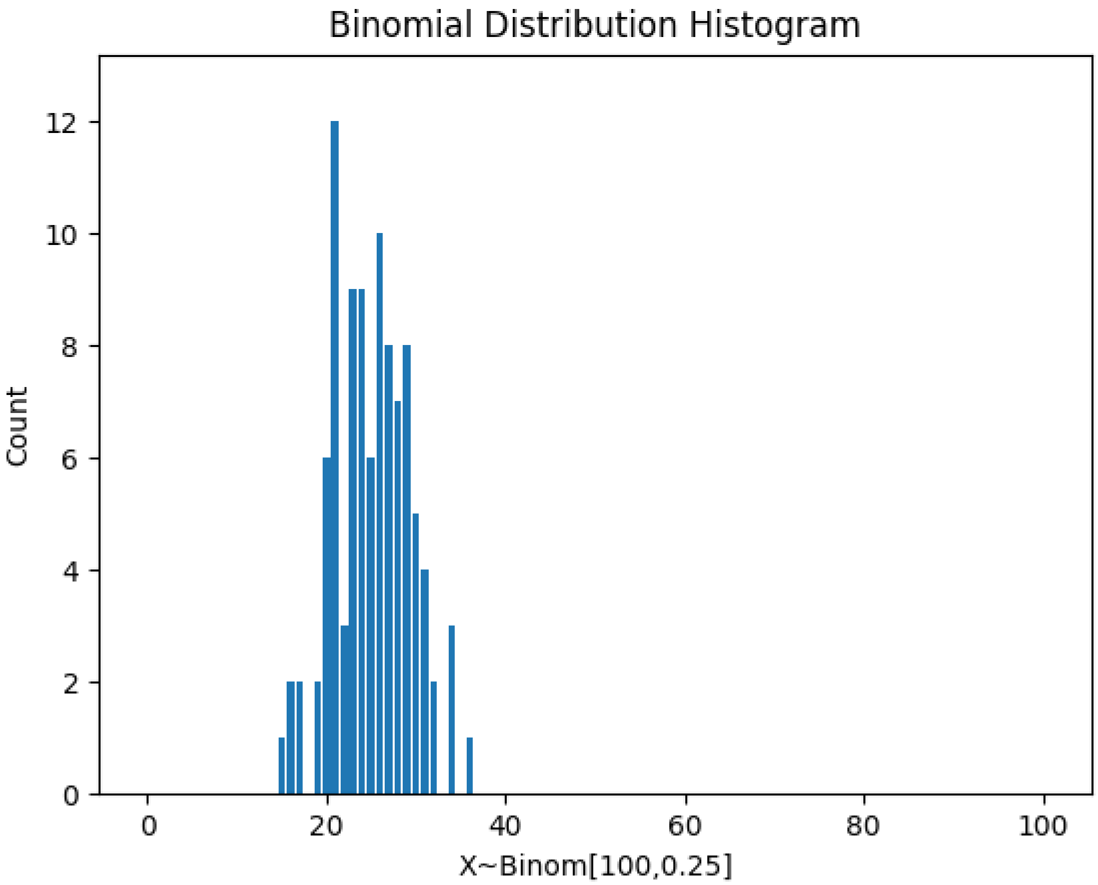


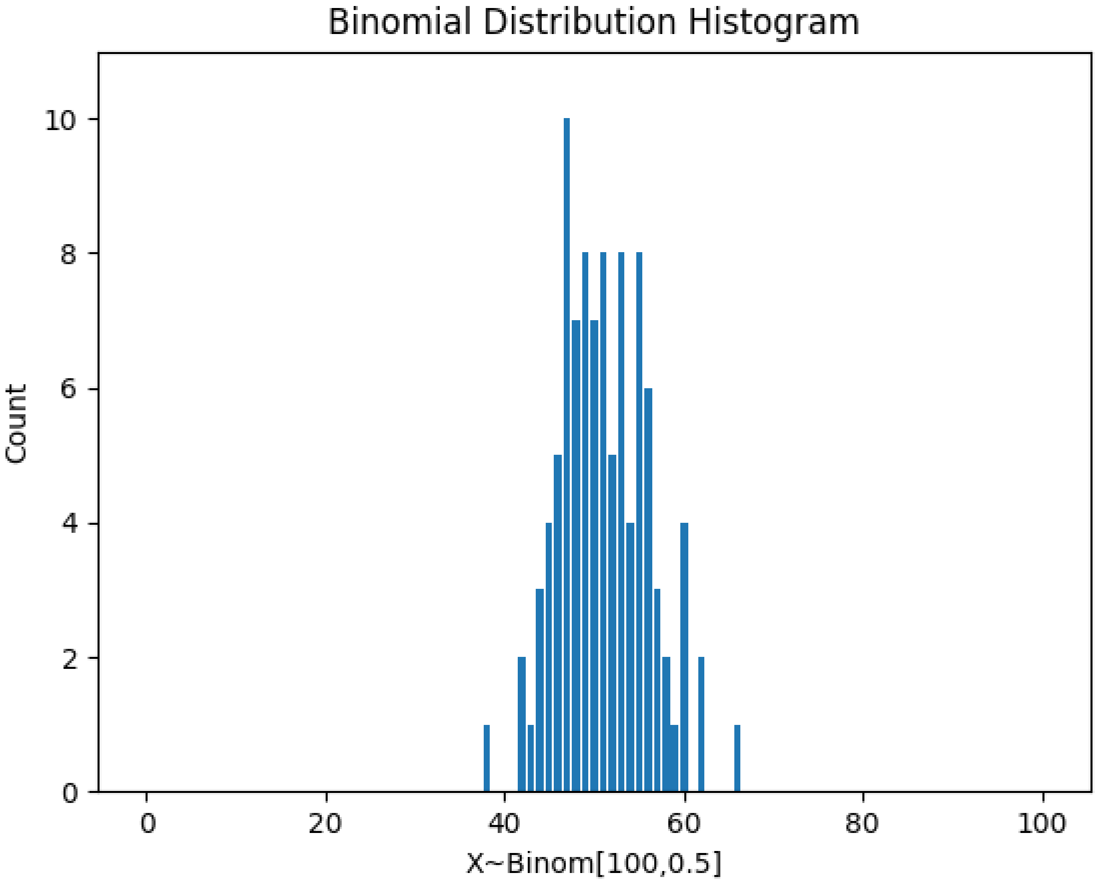
Figure 6: amount=100000

5.3 Rozkład dwumianowy

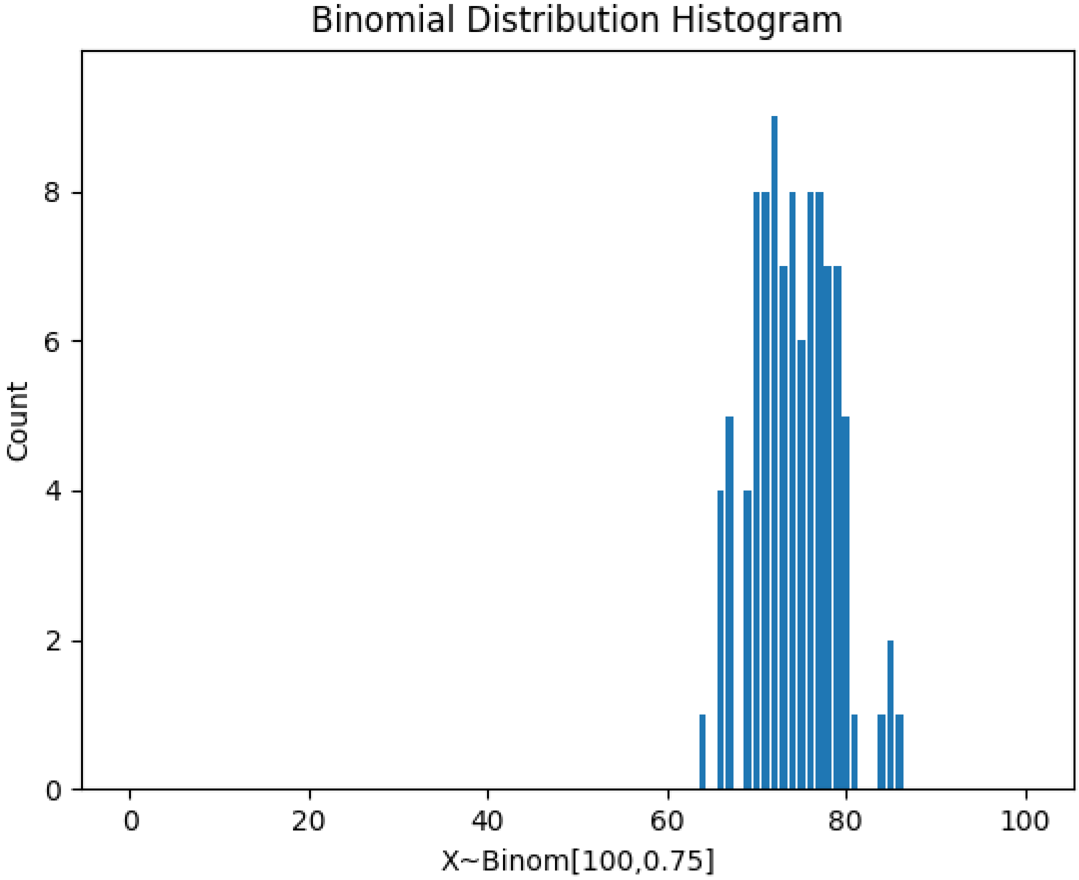
5.3.1 n = 100, p = 0.25



5.3.2 n = 100, p = 0.5


Binomial Distribution Histogram
Count
0
2
4
6
8
10
0
20
40
60
80
100
X~Binom[100,0.5]

5.3.3 n = 100, p = 0.75



5.4 Rozkład Poissona

5.4.1 $\lambda = 1$

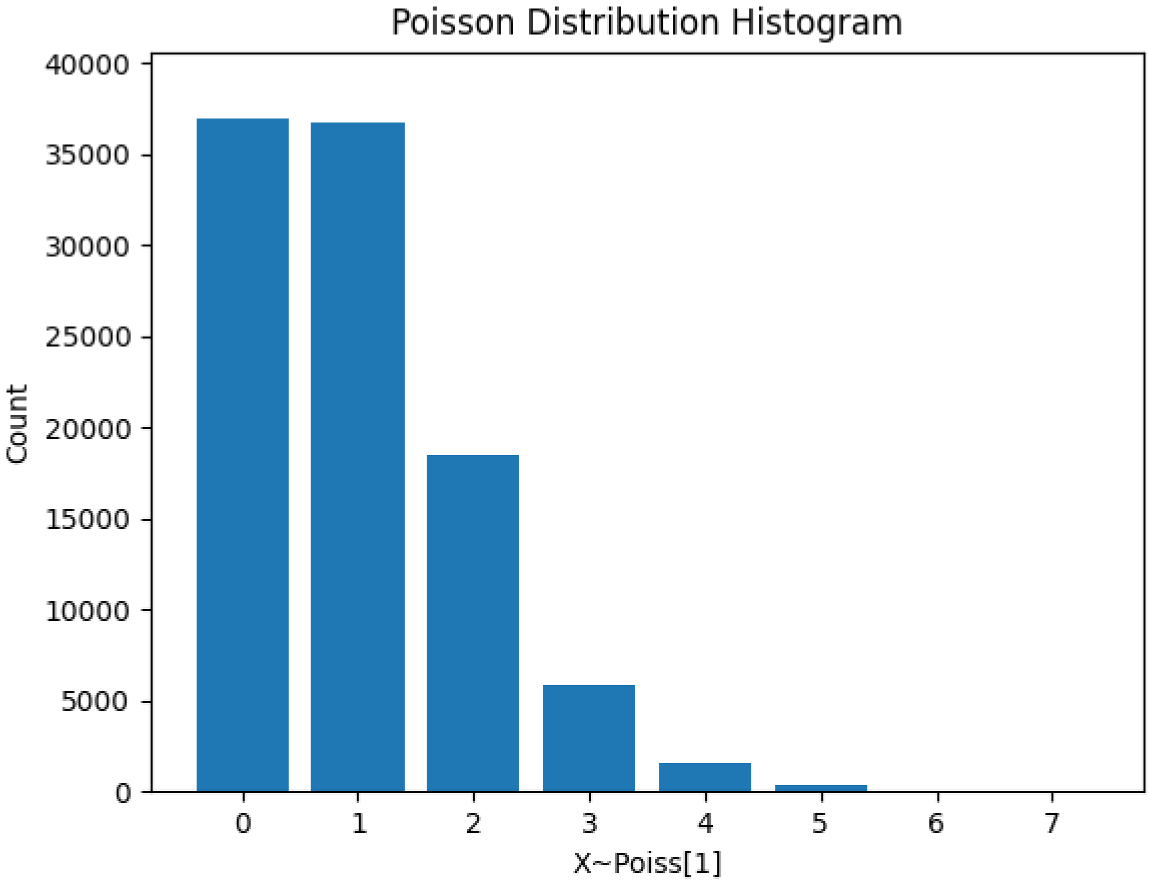


Figure 7: amount=100000

5.4.2 $\lambda = 4$

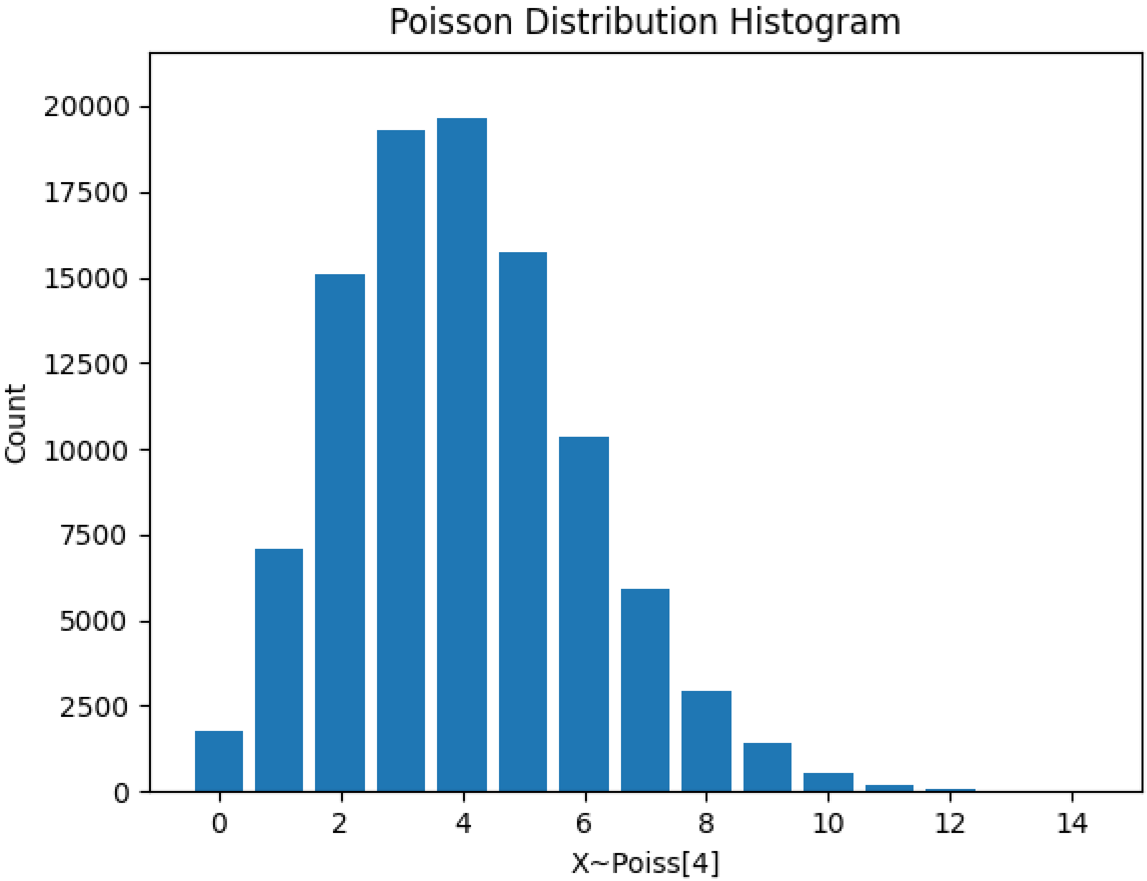


Figure 8: amount=100000

5.4.3 $\lambda = 10$

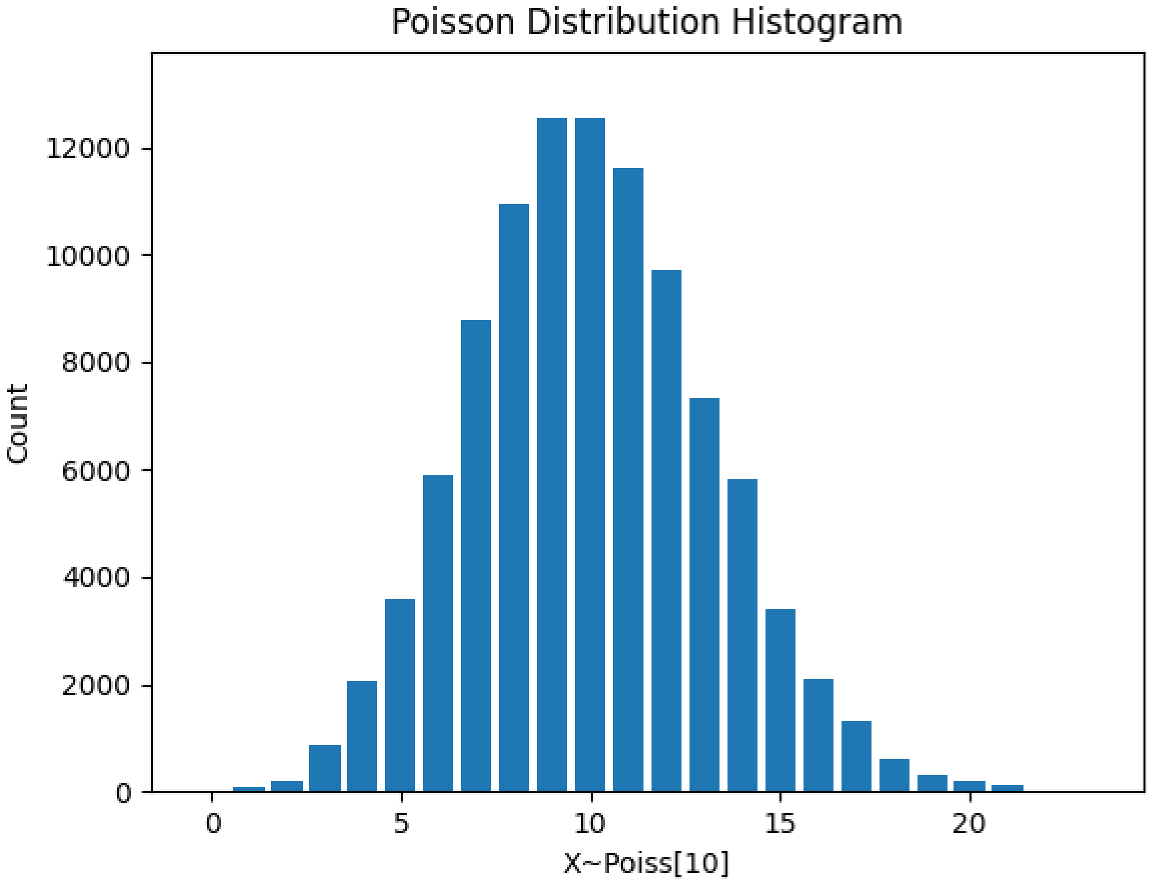


Figure 9: amount=100000

5.5 Rozkład Wykładniczy

5.5.1 $\lambda = 0.5$

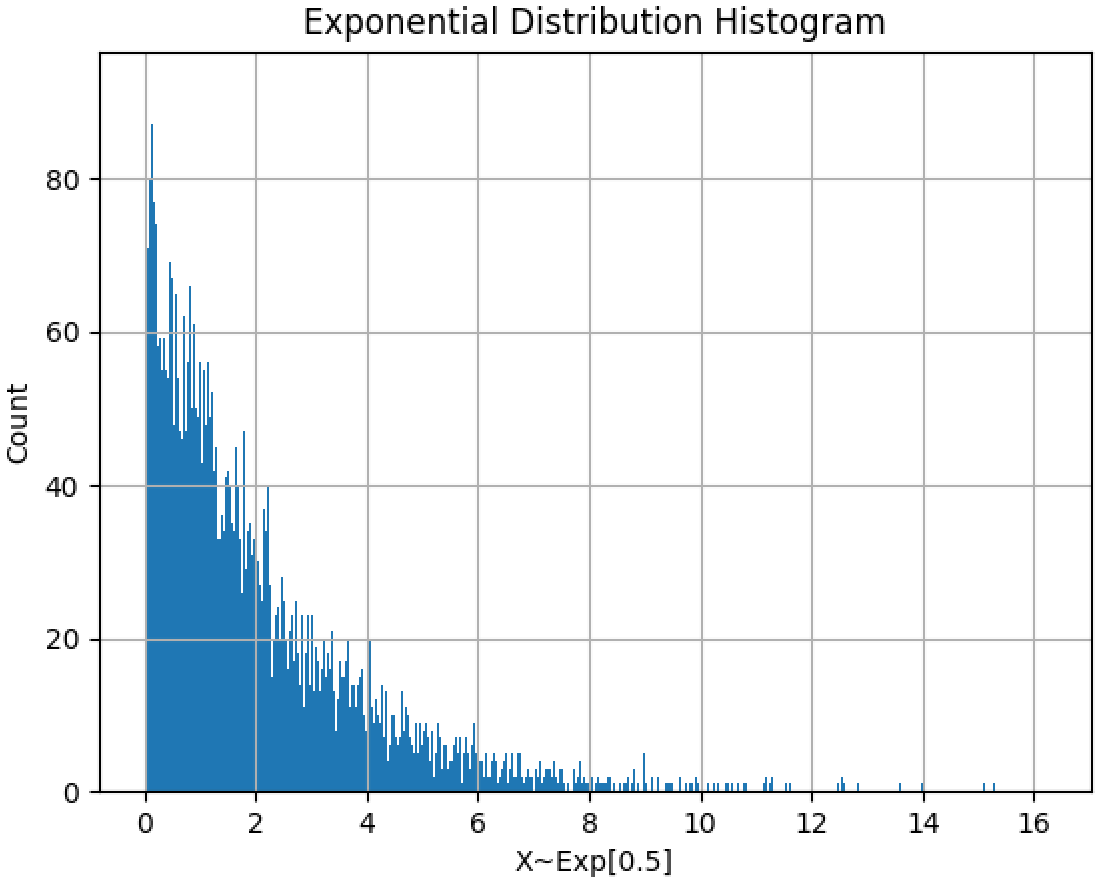


Figure 10: amount=10000

5.5.2 $\lambda = 1$

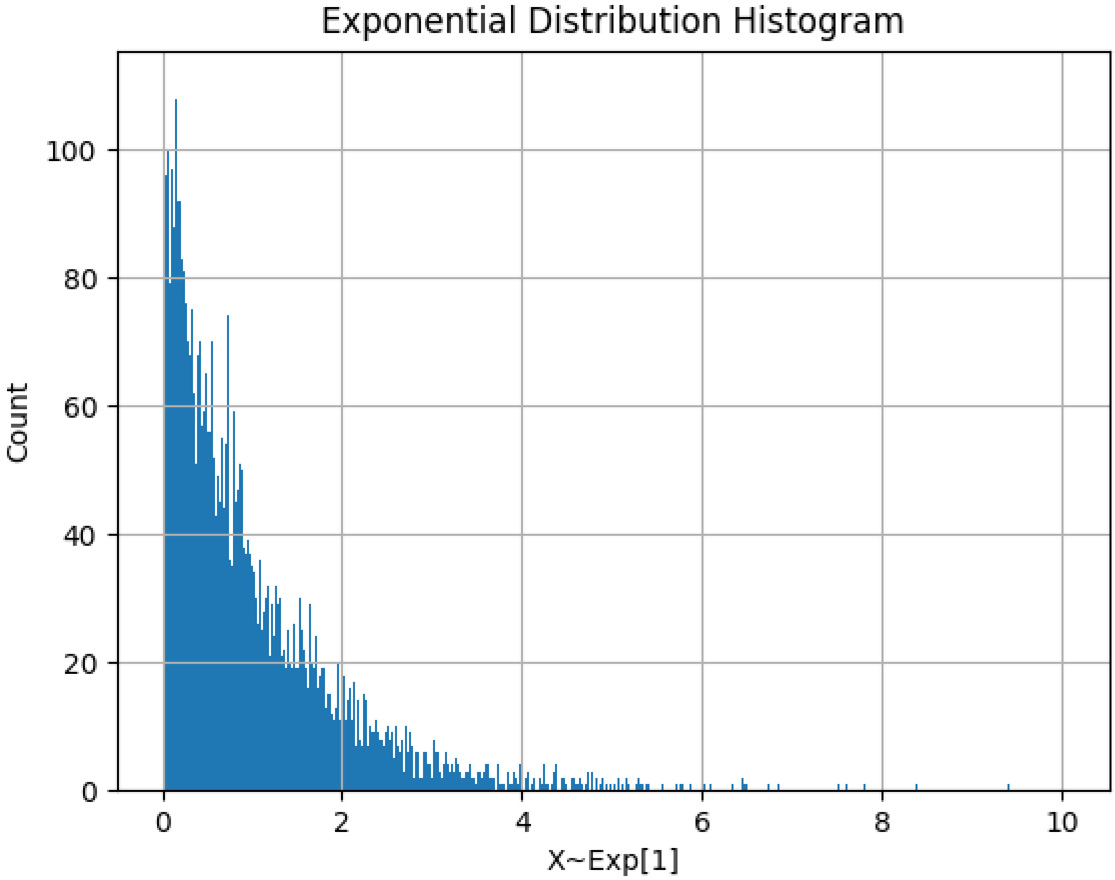


Figure 11: amount=10000

5.5.3 $\lambda = 1.5$

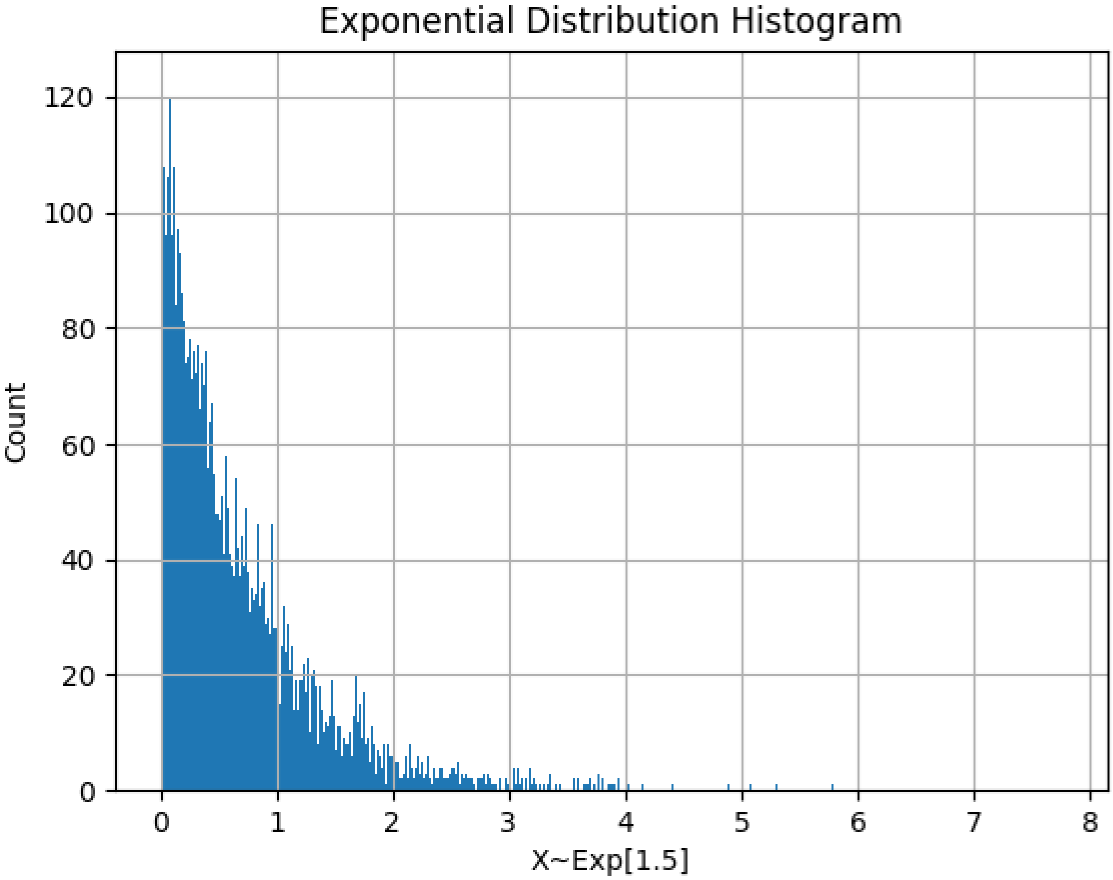


Figure 12: amount=10000

5.6 Rozkład normalny

5.6.1 $\mu = 0$, $\sigma = 0.2$

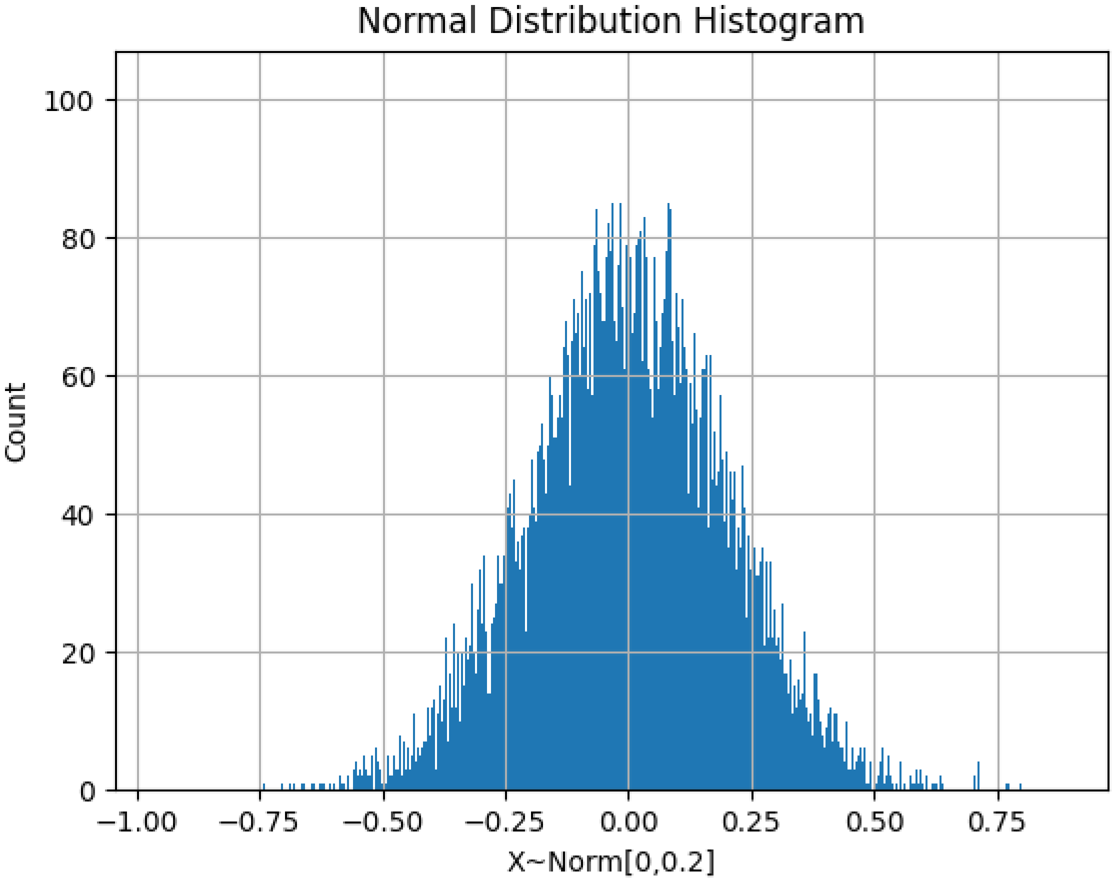


Figure 13: amount=100000

5.6.2 $\mu = 0$, $\sigma = 1$

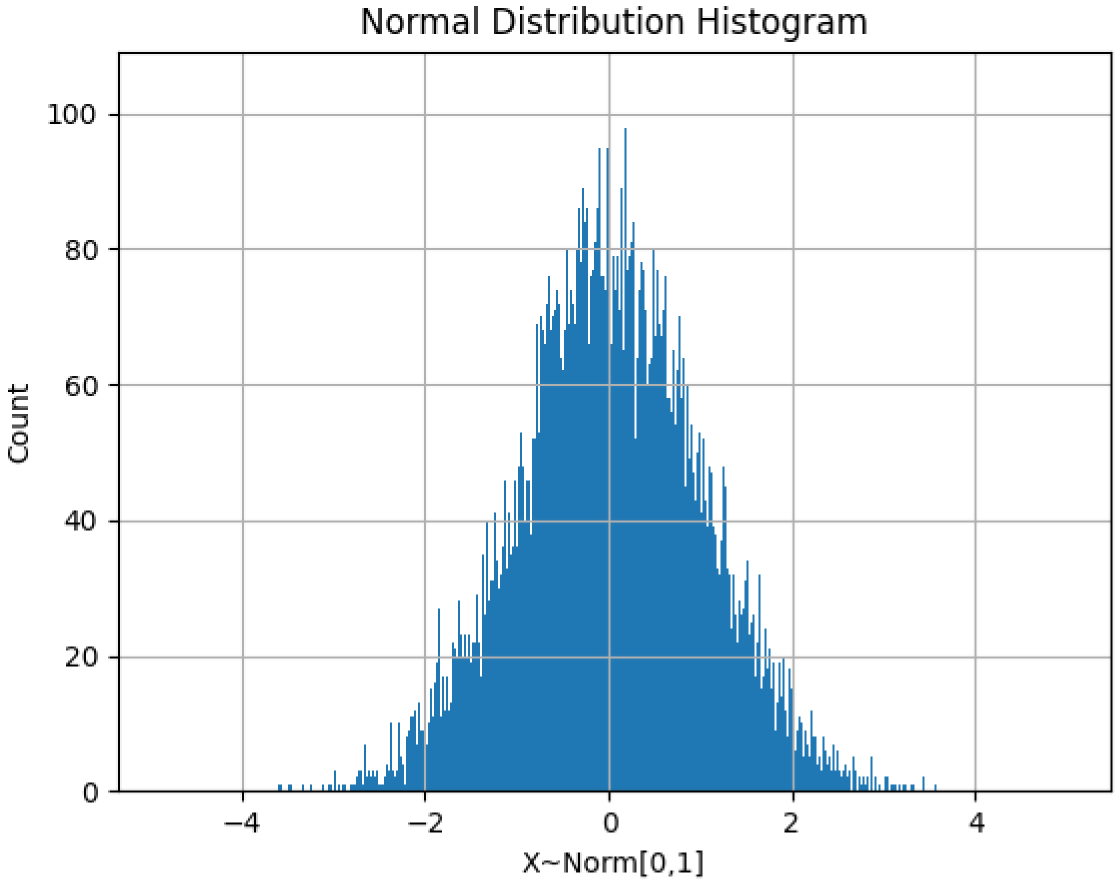


Figure 14: amount=100000

5.6.3 $\mu = 0$, $\sigma = 25$

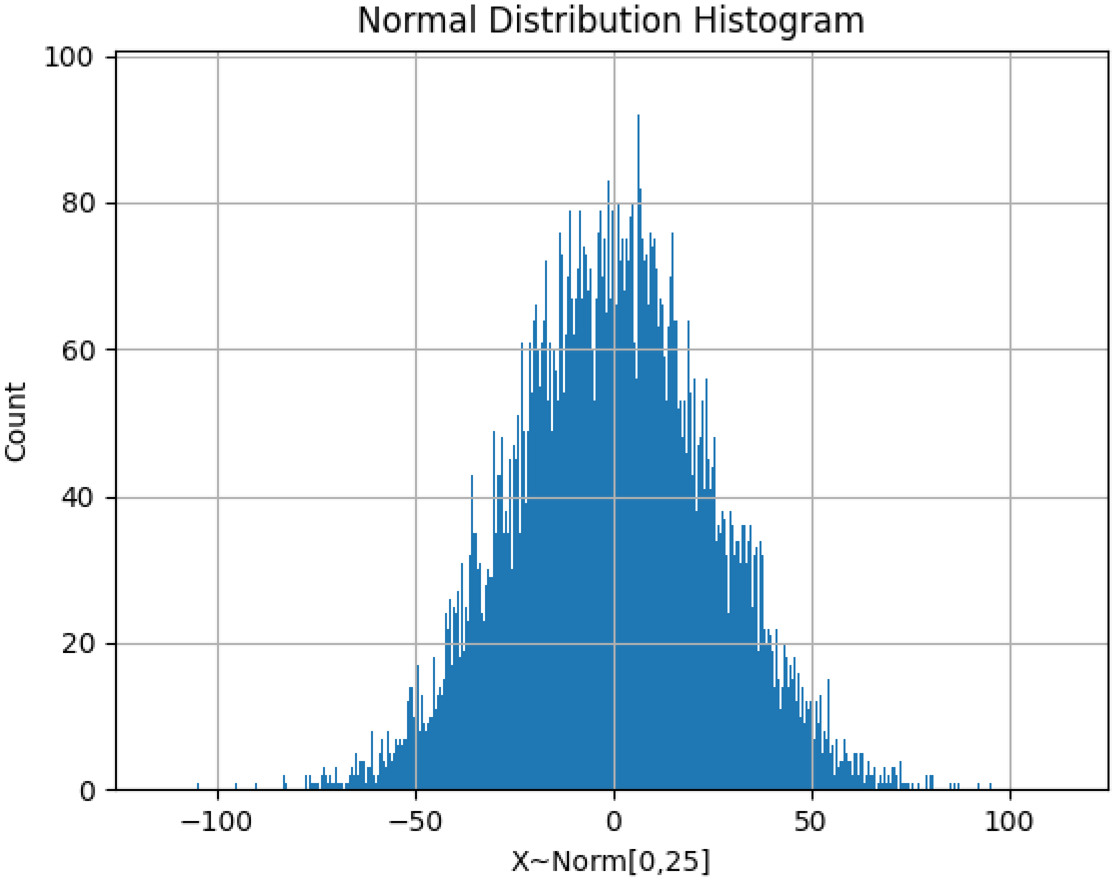


Figure 15: amount=100000

5.6.4 μ = -2, σ = 2

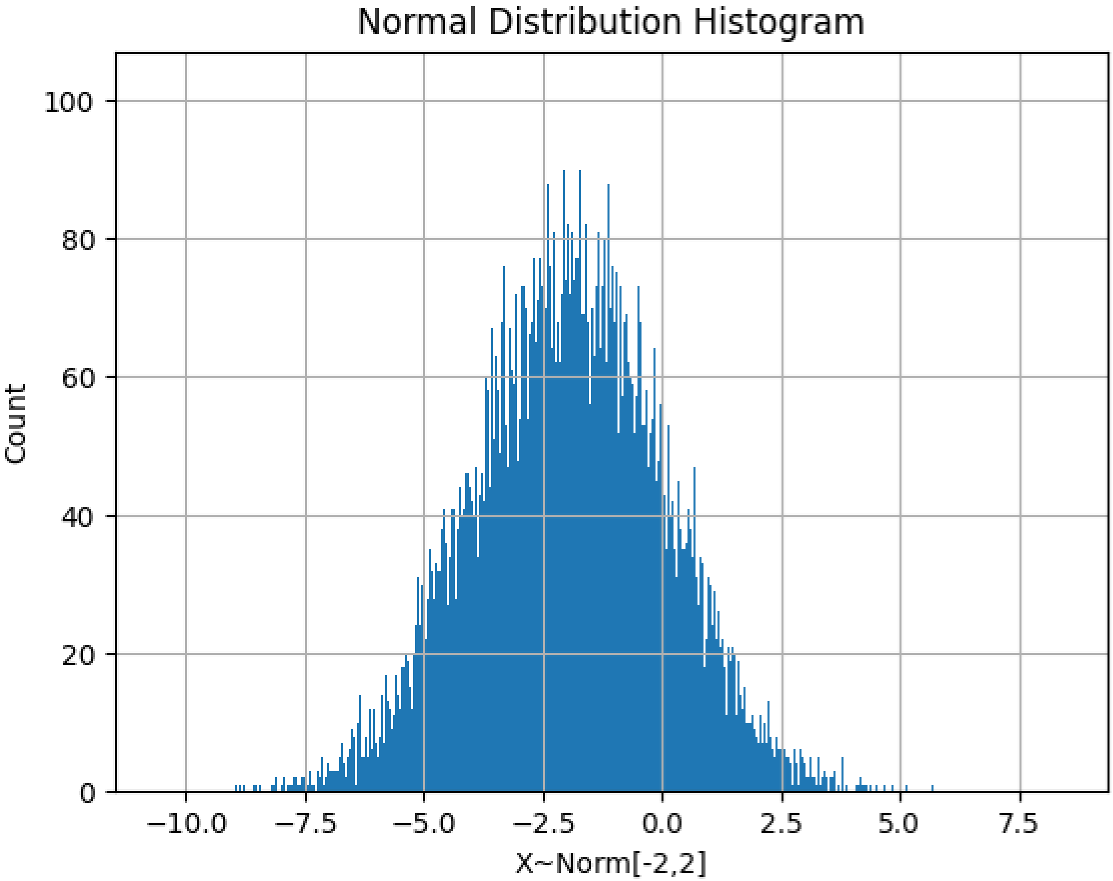


Figure 16: amount=100000