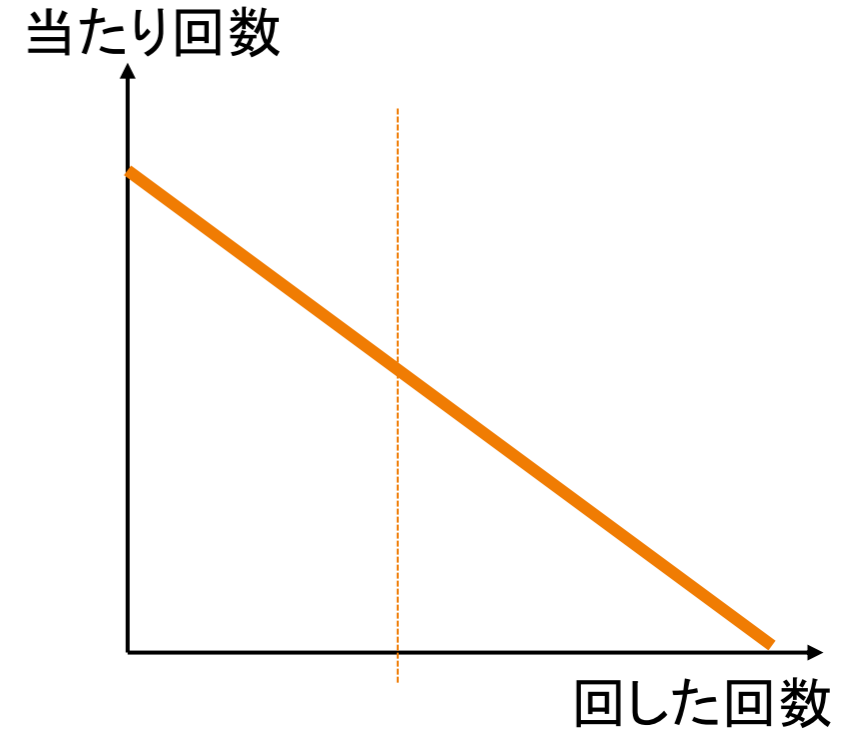
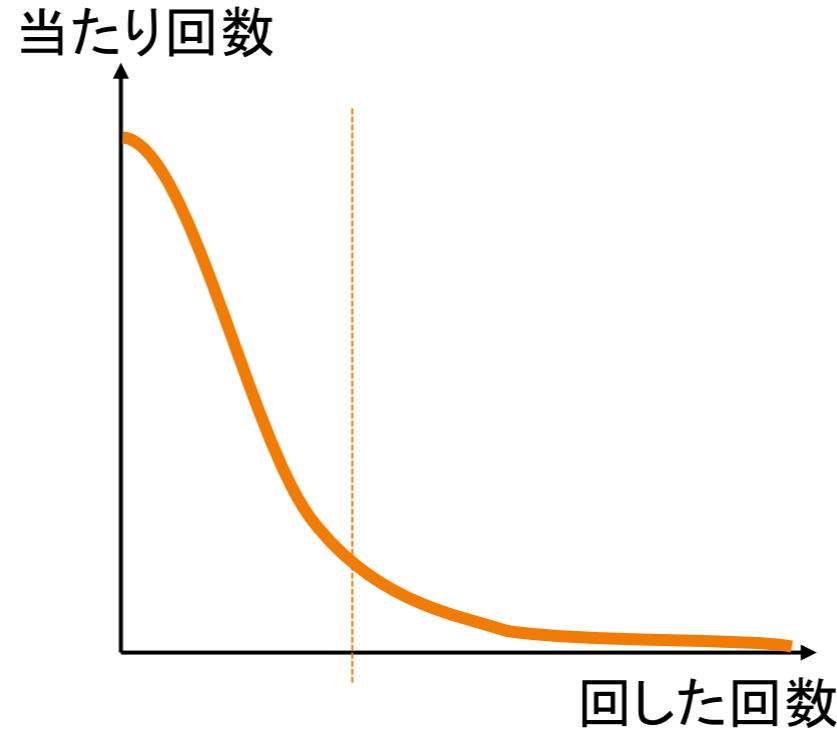
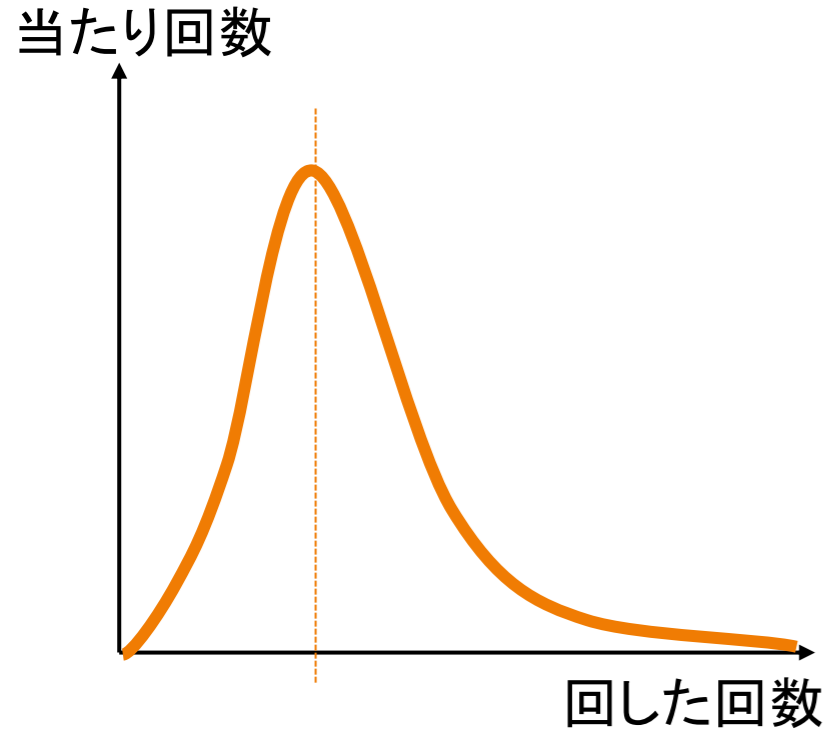


パチンコ簡易シミュレーション

【1000万回まわした当たり分布グラフは？】



- ・ハマり台って、もうすぐ出る？狙い目はあるのかなど

パチンコ機は、中央の入賞口に玉が入るたびに、約1／319の確率で当たり抽選を行う仕組み。



初当たりのみを簡易シミュレーションします。
(確率変動中の当たりは考慮しません。)

1000万回、回した場合に何回転目に当たりが多いかをグラフにした場合を考えます。

入賞口

初当たり確率: 1 / 319.6

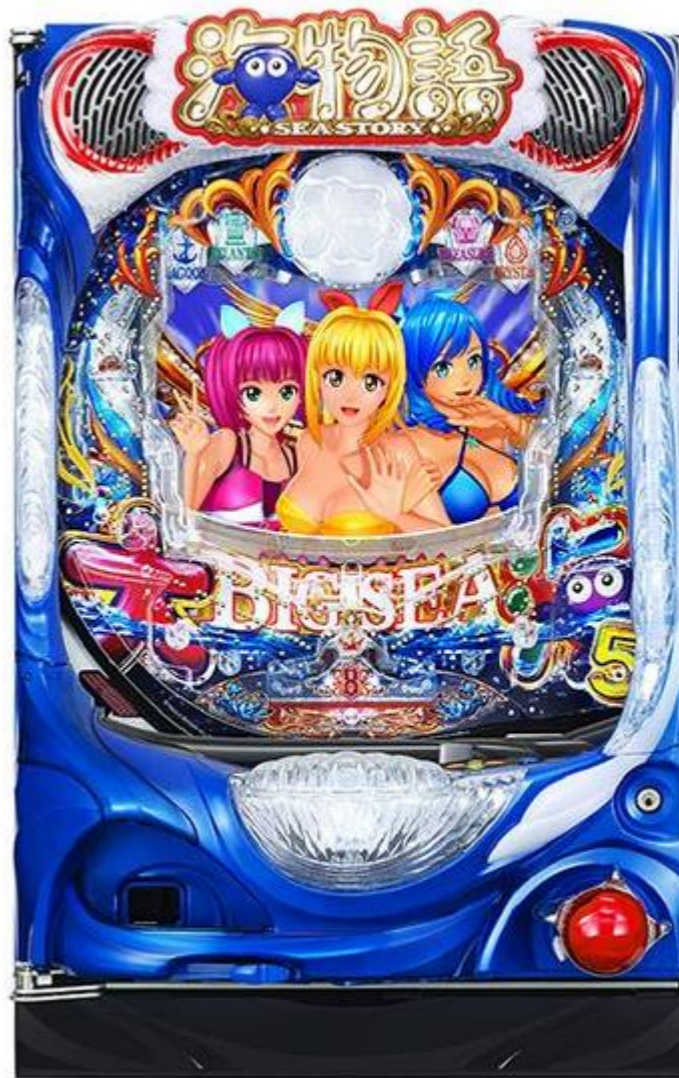
どちらを打ちますか？

1500回転(ハマっている台)



ラッキー、もうそろそろ当たりそう。

0回転(1回転も回っていない台)



どちらに賭(か)けますか

コインを投げて表が出るか裏が出るかを当てるゲーム
(どちらが出るかは50%で表か裏)



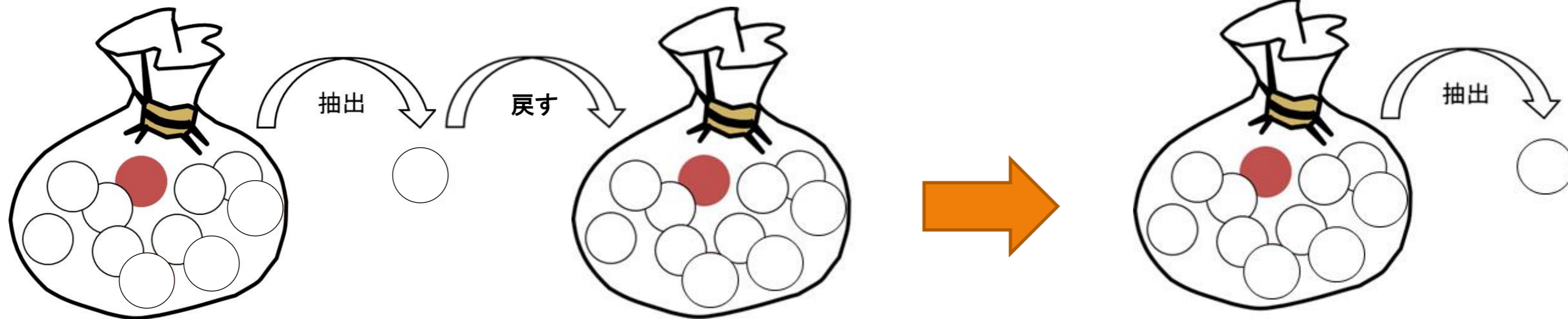
過去7回が全て「表」の場合、次に何が出るかを賭ける

表 1 → 表 2 → 表 3 → 表 4 → 表 5 → 表 6 → 表 7 → 表か裏の
どっちに賭ける？

そろそろ裏が出る？
8回連続する確率は1/256

次は裏が出る方が確率が高い？

1000万回パチンコを回すシミュレーション



319個に1個の当たりが入った袋から抽出します。

確認したら袋に戻します。

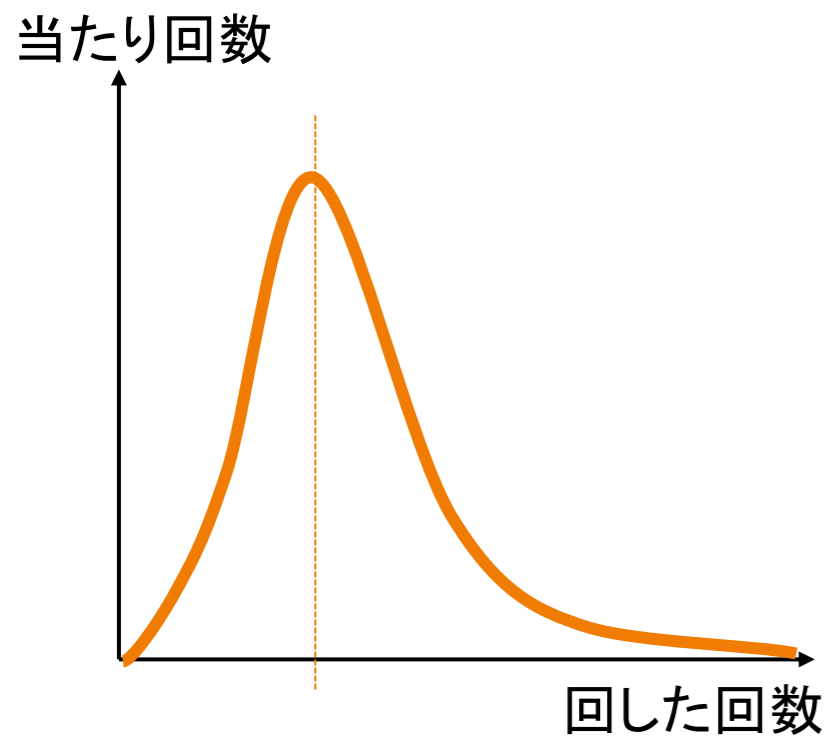
319個に1個の当たりが入った袋から抽出します。

毎回同じ条件で抽出

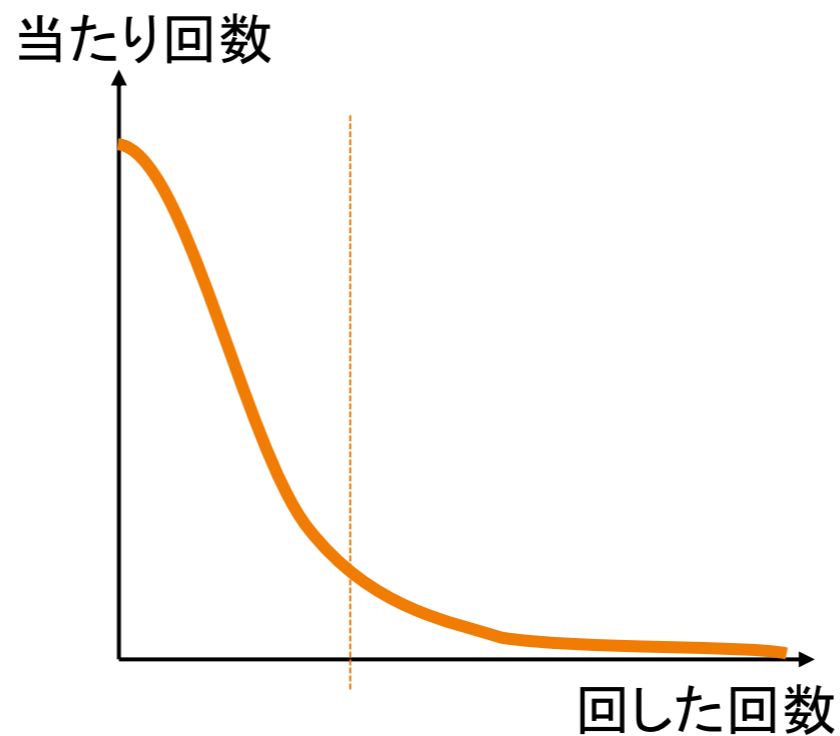
当たりが出るまでの回数をカウントします。あたりが出れば回数は0に戻します。
何回転目に当たりがでるかを1000万回繰り返し分布グラフにします。

1/319の1000万回繰り返した場合、当たり回数と回した回数との関係はどのようなグラフになると思いますか？

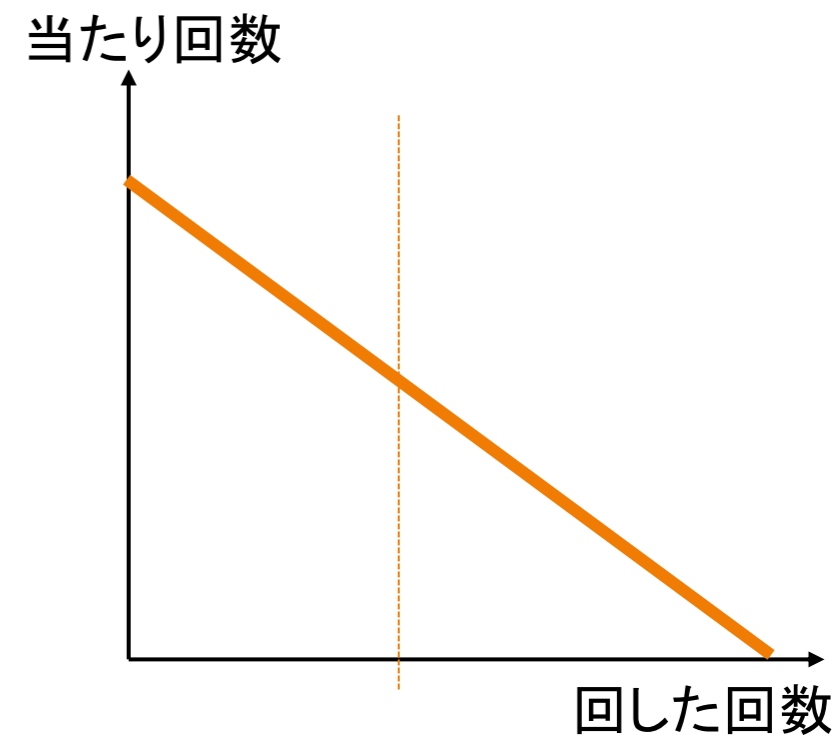
当たり回転数分布グラフ



特定回転数がピーク



回転数毎に当たりが
緩やかに減っていく



回転数毎に当たりが
直線的に減っていく

Pythonプログラム

Hobby-ITサイトにてソース公開

<https://hobby-it.com/>

```
1 import random
2 from google.colab import files
3
4 maxNum = 100000000 # How many bet at this simulation
5 maxBet = 5000 # Max Bet count
6 checkCount = 200 # Check Hit / 200 count
7 dayCount = 2000 # 2000 count / day
8
9 checkCountDiv = maxBet//checkCount # 25
10 dayCountDiv = maxNum//dayCount # How many days (100000000 / 2000)
11
12 hitCount = [0] * (maxBet + 1)
13 hitCheck = [0] * (checkCountDiv) # How many Hit per Check count
14 betCount = [0] * (checkCountDiv) # How many Bet per Cehck count
15 dayCheck = [0] * (dayCountDiv)
16 hit = 0
17
18 for num in range(1, maxNum):
19     hit += 1
20     betCount[((hit-1)//checkCount)] += 1
21     count = random.randint(1,319)
22     if count == 319 or hit == maxBet:
23         if hit == maxBet:
24             print('hit Over maxBet')
25             hitCount[hit] += 1 # How many Hit per Bet Count
26             hitCheck[((hit-1)//checkCount)] += 1 # How many Hit per Check Count
27             dayCheck[((num-1)//dayCount)] += 1 # How many Hit per Day
28             hit = 0
29
30 # Output
31 with open('result1.txt', 'w') as f:
32     for num in range(1, (maxBet+1)):
33         f.write(str(num) + "," + str(hitCount[num]) + '\n')
34 files.download('result1.txt')
35
36 with open('result2.txt', 'w') as f:
37     for num in range(0, checkCountDiv ):
38         f.write(str(num) + "," + str(hitCheck[num]) + "," + str(betCount[num]) + '\n')
39 files.download('result2.txt')
40
41 dayHit = [0] * 30
42 for dNum in range(0, 30):
43     for num in range(0, dayCountDiv ):
44         if dayCheck[num] == dNum:
45             dayHit[dNum] += 1
46 with open('result3.txt', 'w') as f:
47     for num in range(0, 30):
48         f.write(str(num) + "," + str(dayHit[num]) + '\n')
49 files.download('result3.txt')
```

乱数生成(1から319)

1000万回試行を繰り返す
当たるまでの最大回転数(5000)
200回転毎の当たり回数を把握するため
一日あたりの当たり回数を把握するため

配列宣言のため、5000回転まで200回転が25個必要
配列宣言のため、1000万回転まで2000回転が必要数(何日分必要か)

1000万回転を試行
1から319までを乱数で取り出し、319が出た時が当たり
(5000回転まで回した場合は当たりにはしていませんが、ここまでは回ったことはありません。)

ファイル1

回転数毎の当たり回数をファイルに出力

ファイル2

200回転毎の当たり回数をファイルに出力

ファイル3

1日(2000回転)毎の当たり回数をカウントし、ファイルに出力

シミュレーション結果1

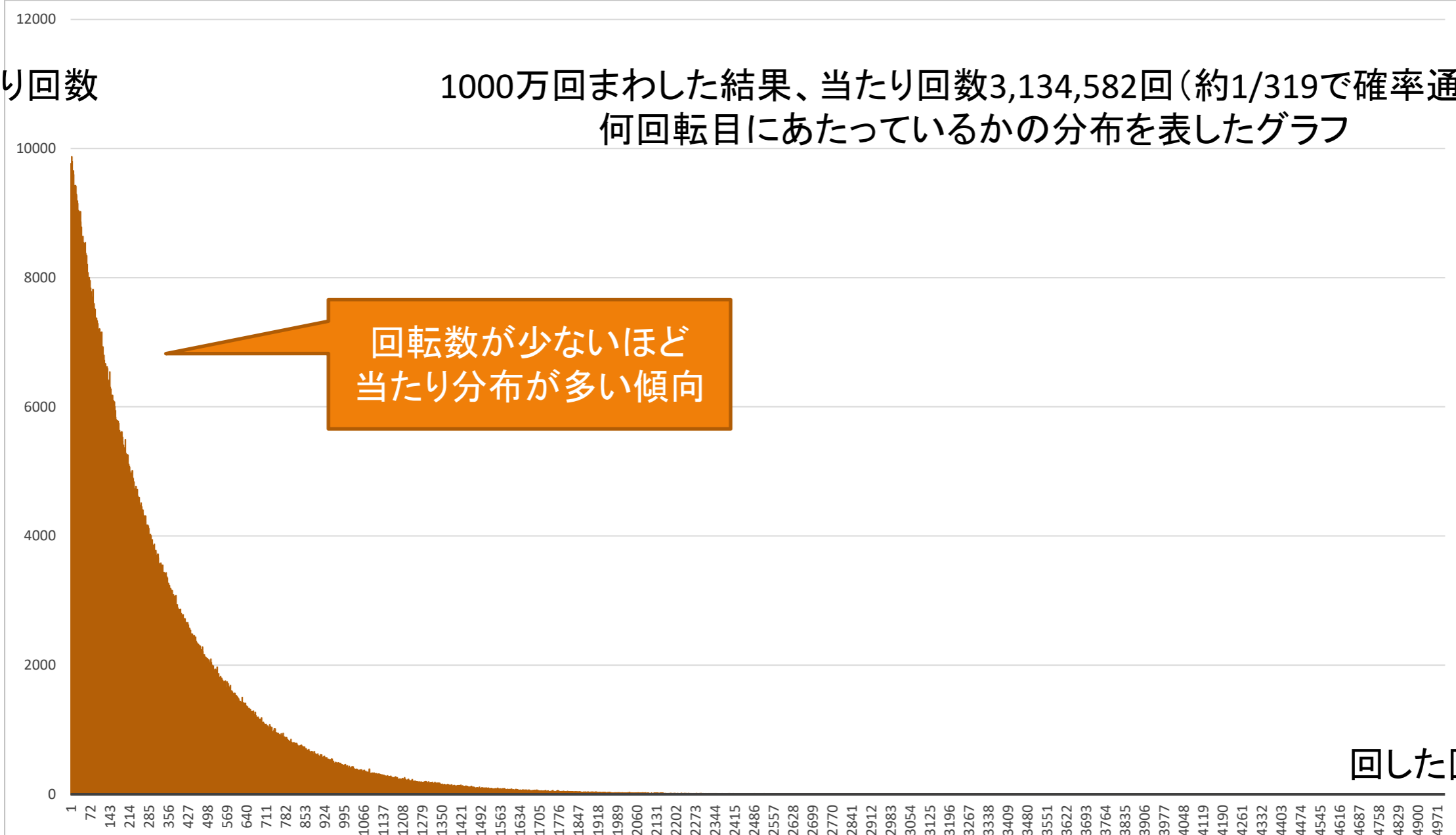
当たり分布グラフ

当たり回数

1000万回まわした結果、当たり回数3,134,582回(約1/319で確率通り)
何回転目にあたっているかの分布を表したグラフ

回転数が少ないほど
当たり分布が多い傾向

回した回数



シミュレーション結果2

当たり分布グラフ(200回転単位)

当たり回数

当たり割合



200回転までに当たる
確率46.6%

200回までの当たり数
全当たり数の割合

回した回数

シミュレーション結果3

回転数	当たり回数	全Bet回数	割合	1 Bet確率
200	1461207	466321685	46.6	0.313347
400	780414	248856794	24.9	0.3136
600	415737	132977917	13.3	0.312636
800	222597	70942543	7.1	0.3137
1000	119213	37811560	3.8	0.315282
1200	63075	20156485	2.0	0.312927
1400	33946	10744644	1.1	0.315934
1600	17947	5707458	0.6	0.314448
1800	9647	3029760	0.3	0.318408
2000	5014	1602734	0.2	0.31284
2200	2714	860844	0.1	0.315272
2400	1423	458067	0.0	0.310653
2600	748	247573	0.0	0.302133
2800	422	135156	0.0	0.312232
3000	213	73258	0.0	0.290753
3200	144	37739	0.0	0.381568
3400	60	16923	0.0	0.354547
3600	29	8863	0.0	0.327203
3800	15	4628	0.0	0.324114
4000	8	2519	0.0	0.317586
4200	3	1480	0.0	0.202703
4400	4	659	0.0	0.60698
4600	0	400	0.0	0
4800	1	292	0.0	0.342466
5000	1	18	0.0	5.555556

当たり割合が多いが、Betしている数も多いため、
確率上、全ての回転数で同じ
1回まわす当たり確率は、1/319 (約0.31347%)

201から400回転で当たる割合は「24.9%」
これは全体からの割合なので、200までの当たりを引いた
残りからの割合は「約46.6%」

200回転まわした場合に当たる確率は「約46.6%」

また、(確率)理論値と合うか計算

1回で当たる確率が1/319で200回チャンスがあるなら
200倍すれば計算できる???


$$(1/319) \times 200 = 0.626959\dots$$


$$= 62.69\%$$

「46.6%」と合っていないし、
当然そんな安易には計算できません！

シミュレーション結果4

よくわからないので、ChatGPTに聞いてみた。

 319個のうち1個の当たりがあります。この抽選を復元抽出を行う場合、200回までに当たりを引く確率はいくらか

 この問題は二項分布を用いて解くことができます。

まず、1回の試行で当たりを引く確率は $1/319$ です。この試行を独立に200回行うと仮定すると、当たりを引く回数 X は二項分布 $B(200, 1/319)$ に従います。ここで、 $B(n, p)$ は試行回数 n と成功確率 p の二項分布を表します。

復元抽出を行っているため、1回の試行で当たりを引いた場合でも次の試行でも再び当たりを引く可能性があることに注意してください。

これらの情報をもとに、200回までに当たりを引く確率は次のように計算できます：

$$\begin{aligned} P(X \geq 1) &= 1 - P(X=0) \\ &= 1 - (1 - 1/319)^{200} \\ &\approx 0.5221 \end{aligned}$$

解答はすごいのに
計算は間違ってる？

従って、200回までに当たりを引く確率は約 0.5221、つまり約 52.21%となります。

$$P(X \geq 1) = 1 - P(X=0)$$

1回以上あたる確率

1から、1度も当たらない確率を引けば良い

$$= 1 - (1 - 1/319)^{200}$$

1度も当たらない確率は、1から1/319引く(318/319でも同じ)

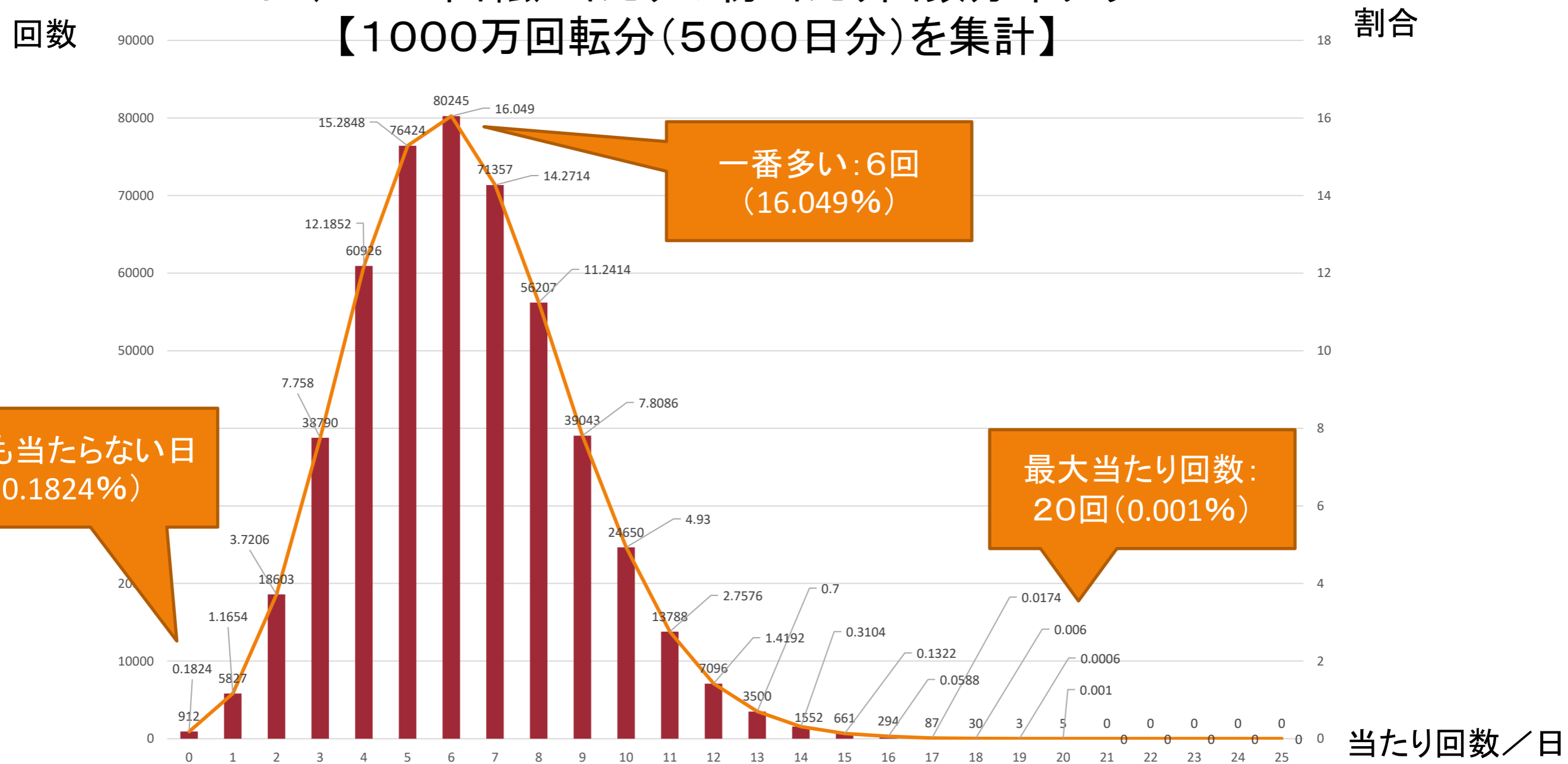
$$= 0.466312 \dots$$

$$\doteq 46.6\%$$

理論計算値が合致したので
シミュレーションはやっぱり正しい！

シミュレーション結果5

1日(2000回転)当たりの初当たり回数分布グラフ 【1000万回転分(5000日分)を集計】



考察とまとめ

- 一番当たる回転数は存在するか？
存在しない。少ない回転数ほど、当たり回数の分布は多いが、確率が高いわけではない。
1から200までで、当たり全体の約半分(46.6%)となった。
- ハマリ台か、当たっている台を狙う価値はあるか？
過去の状態は関係なく、毎回319分の1で抽選を行うため、理論上、確率は同じ。
どの台も確率上は全て同じなので、どの台を打っても全ては運しだい。
- 台移動するか、粘って打つか、どちらが良いか？
理論上、確率は同じなので、どちらが良いとは言えない。
- どれくらいハマる可能性があるか？
4000以上(5000近く)ハマる回数があったので、2日間当たりなしもありえる。
ただし、2200回転までで当たるのは99.9%
- 回る台は本当に良いのか？
確実に良い。回す回数で確率があがる。一回当たりのコストが安い方が良い。

シミュレーションの要望などあれば、お願いします！

コメント欄やサイトの問い合わせでお待ちしています。

(電子工作なども含めてIT関連何でもOKです。)