A tutorial on the mc-tk toolkit

The mc-tk toolkit includes three sub-modules(1). The next sections will demonstrate all the funcitons in each module.

Module	Function	Description
mc. experiments	pi()	Perform Buffon's needle experiment to estimate π.
	parcel()	Simulate a bi-directional parcel passing game.
	dices()	Estimate the probabilities of various dice combinations.
	prisoners()	The famous locker puzzle(100-prisoner quiz). This function will prove that the survival chance limit is $1 - ln2$ when n approaches $+\infty$.
	prisoners _limit()	
	galton _board()	Use the classic Galton board experiment to produce a binomial distribution.
	paper _clips()	Use the paper clip experiment to produce a Zipf distribution.
	sudden _death()	This function simulates a sudden death game to produce the exponential distribution.
mc. distributions	poisson()	This function will demonstrate that Poisson is a limit distribution of *b(n,p)* when n is large, and p is small.
	benford()	Verify Benford's law using real-life datasets, including the stock market data, international trade data, and the Fibonacci series.
mc. samplings	clt()	Using various underlying distributions to verify the central limit theorem. This function provides the following underlying distributions.
		'uniform' - a uniform distribution U(-1,1).
		'expon' - an exponential distribution Expon(1).
		'poisson' - poisson distribution $\pi(1)$.
		'coin' - Bernoulli distribution with p = 0.5.
		'tampered_coin' - PMF:{0:0.2,1:0.8}, i.e., head more likely than tail.
		'dice' - PMF:{1:1/6,2:1/6,3:1/6,4:1/6,5:1/6,6:1/6}.
		'tampered_dice' - PMF: {1:0.1,2:0.1,3:0.1,4:0.1,5:0.1,6:0.5},i.e., 6 is more likely.
	t_stat()	This function constructs an r.v. (random variable) following the t distribution.
	chisq_gof _stat()	Verify the statistic used in Pearson's Chi-Square Goodness-of-Fit test follows the χ^2 distribution.
	fk_stat()	Verify the Fligner-Killeen Test statistic(FK) follows the χ^2 distribution.
	bartlett _stat()	Verify the Bartlett's test statistic follows the χ^2 distribution.
	anova_stat()	Verify the statistic of ANOVA follows the F distribution.
	kw_stat()	Verify the Kruskal-Wallis test statistic (H) is a χ^2 r.v.
	sign_test _stat()	For the sign test (medium test), verify its N- and N+ statistics both follow *b(n,1/2)*.
	cochrane _q_stat()	Verify the statistic T in Cochrane-Q test follows the χ^2 distribution.
	hotelling _t2_stat()	Verify the T^2 statistic from two multivariate Gaussian populations follows the Hotelling's T^2 distribution.

Table 1: An overview of the software modules and functions

1. Functions for classical MC experiments

The mc.experiments module introduces some simulation experiments of classical probability problems.

1.1. Estimate π

The pi() function provides a typical experiment to estimate π. It is Buffon's classical needle experiment [1]. Buffon proved that the probability of the stick intersecting any parallel lines is $p = 2l/\pi d$. l is the needle length, and d is the distance between parallel lines. By calling the pi() function, we can estimate $\hat{\pi} = 2l/df$. f is the frequency; when the stick number is large enough, the estimate will approach the real π.

```
pi(N=1000000,d=4,l=1)
# N : the number of needles.
# d : the distance between two parallel lines
#     in the Buffon's needle problem.
# l : the length of an arbitrary needle cast
#     in the Buffon's needle problem.
```
result : PI = 3.141441716926069

1.2. The Parcel-Passing Game

The parcel() function is designed to solve the probability problems in the bi-directional parcel-passing game. The standard setting for this game is as follows. Five players (A, B, C, D, E) form a circle. In each round, the parcel holder can pass the parcel either to his/her left or to the right. The question is: after ten rounds, what is the probability of the parcel returning to the starter player?

```
parcel(N=100000,num_players=5,num_ops=10)
# N : how many MC experiments to run.
# num_players : the number of players.
# num_ops : the number of times the parcel is passed
#           during each experiment.
```
result : p = 0.247

The theoretical solution to this problem is $P = \frac{C_{10}^0+C_{10}^0+C_{10}^5}{2^{10}} = \frac{2+C_{10}^5}{2^{10}} = 0.248$. The nominator $(C_{10}^0+C_{10}^{10}+C_{10}^5)$ stands for the qualified cases, i.e., out of 10 game rounds, 0, 10, and 5 times are passed to the left. The denominator is the number of all possible cases. The parcel() function simulates this game, and by using 100,000 simulations, the observed frequency is about 0.247, which is very close to the theoretical value. The parcel() function also allows users to set different player numbers and game rounds.

1.3. Dice Gambling

The dice() function simulates the dice game in the Japanese manga "Kaiji: The Ultimate Gambler." In this game, the player will toss three dice. Based on the outcome, they will get different rewards (Table 2). The question is: what is the mathematical expectation of the reward? In other words, what will be a fair price for each game if you are the casino boss?

Table 3 is the result after 10,000 MC simulations. The observed frequencies are very close to the theoretical PMF (probability mass function). Furthermore, we can calculate the estimated math expectation of the reward is $\widehat{E(x)} = \sum f_k X_k = 1.962$.

```
dices(N=10000)
# N : how many MC experiments to run.
```

Result	Reward
1 2 3	16
4 5 6	16
Triple, or Three of a Kind (e.g., 4 4 4)	8
Pair(e.g., 4 4 3)	2
Single(e.g., 1 2 6)	0

Table 2: Score table for different results

	123	456	Triple	Pair	Single
Experimental Frequencies (f)	0.0287	0.0284	0.0273	0.415	0.501
Theoretical PMF(p)	0.0278	0.0278	0.0278	0.417	0.499

Table 3: Results of 10000 MC simulations

1.4. The locker puzzle

The "hundred-prisoner puzzle" or "the locker puzzle" was first addressed by Danish scientist Peter Bro Miltersen [2] [3]. In this puzzle, there are 100 lockers containing No.1 to No.100. In each round, one prisoner will open 50 lockers. The game will continue if his/her number is found inside any of the opened lockers. Otherwise, the game is over, and all prisoners will be executed. The prisoners cannot communicate with each other during the game. What are the best strategy and the highest survival probability?

With no strategy (becomes a repeated Bernoulli experiment), the survival probability will be $(\frac{1}{2})^{100}$, which is virtually 0. According to the authors, the best strategy is the "circular chain", i.e., the prisoner first opens the locker of his or her number, then opens the locker whose number is inside the last locker. With this strategy, the survival probability equals the probability of creating circular chains no longer than 50. This probability is $p = 1 - \frac{1}{100!}\sum_{l=51}^{100}(\frac{1}{l} \times 100!) = 1 - \sum_{l=51}^{100}\frac{1}{l} = 1 - 0.688 = 0.312$. Furthermore, if we increase the total prisoner number, we can prove that this probability will converge to $1 - ln2$ (0.307).

The prisoners() function can simulate this experiment, and users can get the survival chance plot against different prisoner numbers(Figure 1).

```
prisoners(n=100,N=2000)
# n : the number of prisoners.
# N : how many MC experiments to run.
result : p = 0.3116
```

```
prisoners_limit(ns=[250,500,750,1000,1250,1500,1750,2000],
    N = 1000)
# ns : the number of prisoners to be tested each time.
# N : the number of MC experiments performed for each n.
```

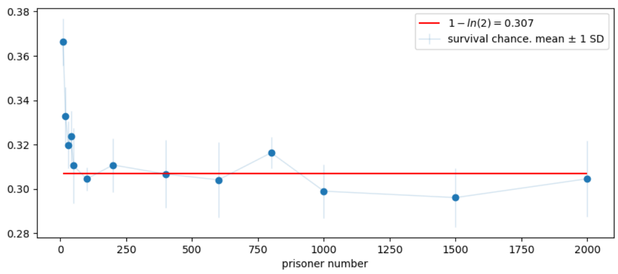


Figure 1: Survival chance against prisoner number (n). When n is large enough, this chance will approach 1-ln2.

1.5. Galton Board

The binomial distribution originates from the repeated independent Bernoulli trials. Its PMF (probability mass function) is defined as $P(x = k) = C_n^k p^k (1-p)^{n-k}, k = 0, 1, ..., n$, n is the trial number. p is the probability of interest event in each Bernoulli trial. The Galton board, a.k.a. quincunx or bean machine, is a historical device to demonstrate the binomial distribution. mc-tk provides a binom() function to simulate this experiment (Figure 2).

```
galton_board(num_layers=20, N=5000)
# num_layers : the number of nail plate layers.
# N : how many MC experiments to run.
```

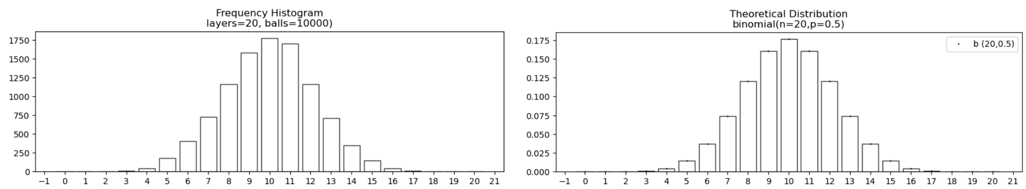


Figure 2: Use the Galton board MC experiment (10000 balls, 20 layers) to reproduce the binomial distribution.

1.6. Survival Game

The exponential distribution describes a continuous random variable with the following PDF (probability density function):

$$p(x) = \begin{cases} \theta e^{-\theta x}, & x > 0 \\ 0, & x \le 0 \end{cases} \quad (\theta > 0) \tag{1}$$

In mc-tk, we define a survival game to illustrate the underlying mechanism of the exponential distribution. Because the sudden death game can approximate many real-life accidents or electronic component failures (e.g., capacity breakdown or LCD pixel defect), the resulting exponential distribution can be used in survival analysis and lifespan estimation.

In each round of this survival game, the test subject (player) is faced with a very low sudden death probability (p). In Figure 3, we choose p = 0.001 and simulate 10,000 MC rounds. The generated histogram is very close to the exponential distribution. This function can be used to illustrate the generation mechanism of the exponential distribution.

```
sudden_death(num_rounds=1000,p=0.01,N=10000)
# num_rounds : the rounds of survival games.
# p : the probability of sudden death/failure/accident
#     in each round.
# N : how many MC experiments to run.
```

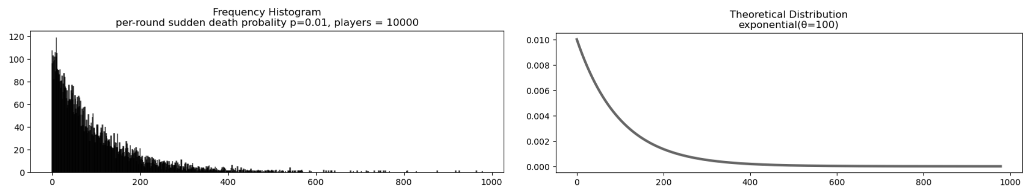


Figure 3: The observed frequency histogram of the survival game and the corresponding theoretical exponential distribution.

1.7. Paper Clip Experiment

The Zipf law was proposed by George Kingsley Zipf in 1949 from his linguistic research [4]. The Zipf law says that a word's frequency in natural language is inversely proportional to its rank. This means only a few words are frequently used, and most are seldom used. This phenomenon is known as the 80/20 law, the long tail distribution, or the Pareto principle. The PDF of the Zipf distribution is as follows.

$$p(k; a) = \frac{1}{\zeta(a) k^a} \tag{2}$$

$k \ge 1$,$a > 1$. a is the shape parameter. ζ is the Riemann zeta function, which is defined as:

$$\zeta(x) = \sum_{n=1}^{\infty}(\frac{1}{n^x}) = \sum_{n=1}^{\infty} n^{-x} = \frac{1}{1^x} + \frac{1}{2^x} + \frac{1}{3^x} + \frac{1}{4^x} + ... \tag{3}$$

It has many interesting properties, e.g.,

$$\zeta(1) = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + ... = +\infty \tag{4}$$

$$\zeta(2) = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + ... = \frac{\pi^2}{6} \tag{5}$$

$$\zeta(-1) = 1 + 2 + 3 + 4 + ... = -\frac{1}{12} \tag{6}$$

$$\zeta(-2) = 1^2 + 2^2 + 3^2 + 4^2 + ... = 0 \tag{7}$$

The MC experiment related to the Zipf distribution is the paper clip experiment. Each time, we drew two clips from a pipe of paper clips. The picked clips are connected and then put back. After enough rounds, the clip chains of different lengths will obey the Zipf distribution. Users may call the paper_clips() function to simulate this experiment (Figure 4).

```
zipf(num_rounds=10000,num_clips=16000,verbose=False)
# num_rounds : the number of rounds in the paper clip
#              experiment.
# num_clips : the total number of paper clips.
#             It should always be greater than
#             [num_rounds].
```

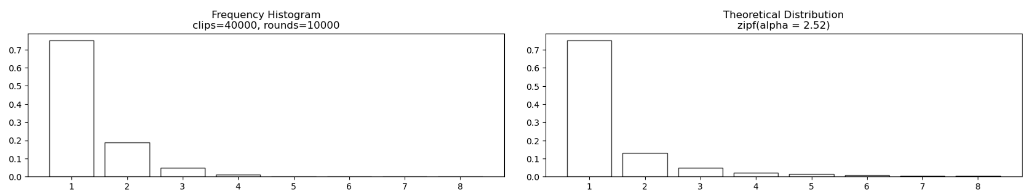


Figure 4: Use the paper clip MC experiment to generate the Zipf distribution.

2. Functions for common distributions

The mc.distributions module uses MC experiment to generate specific distributions. The observed MC result and theoretical distribution PDF/PMF are provided side by side.

2.1. Poisson Distribution

The Poisson distribution has the following PMF:$P(X = k) = \frac{\lambda^k}{k!}e^{-\lambda}, k = 0, 1, ..., \lambda > 0$. Many daily-life events follow the Poisson distribution, e.g., the car accidents that happen every day, the patient visits in the emergency department, etc. The Poisson distribution can be seen as a particular case of the binomial distribution when p is very low and n is very large. Figure 5 demonstrates the poisson() function in mc-tk. In each MC round, a large sample size (n = 10000) is used, and each individual is faced with an extremely low accident probability (p = 0.0001). By simulating 100000 MC rounds, we can see that the total number of accidents follows a perfect Poisson distribution.

```
poisson(n=10000,p=0.0001,N=100000)
# n : sample size.
# p : the extremely low probability of occurrence of accidents
#     that each sample is facing.
# N : how many MC experiments to run.
```

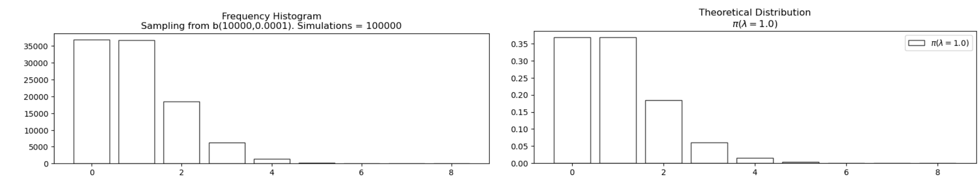


Figure 5: The observed frequency histogram of 100,000 random samples drawn from b(10000, 0.0001) and the corresponding theoretical Poisson distribution.

2.2. Benford Distribution

The Benford law, a.k.a. the Newcomb-Benford law or the first-digit law, describes the PMF of leading digits in many real-life financial and social data [5]. In essence, the natural or social processes that follow the power laws (very common) often demonstrate this distribution. In financial audits, it is often used to check faked or manipulated data. The Benford PMF is as follows (Table 4).

leading digit	1	2	3	4	5	6	7	8	9
p	30.1%	17.6%	12.5%	9.7%	7.9%	6.7%	5.8%	5.1%	4.6%

Table 4: Leading digit PMF

The benford() function provides three examples to verify the Benford law (Figure 6). The first example uses 20-year trading volume data of AAPL (Apple Inc.). The second example uses United Nations' international trading data. The last example uses the Fibonacci series.

```
benford(data="stock",N=1000)
# data : data set/experiment to be used.
#     'stock' - use 20-year stock trading volume
#               data of Apple Inc. (AAPL)
#     'trade' - use annual trade data from various
#               countries. https://comtrade.un.org/data/mbs
#     'fibonacci' - use the top-N fibonacci series.
# N : how many MC experiments to run.
```

According to Figure 6, all the examples fit well against the theoretical Benford distribution. We can use the Fibonacci series to explain the Benford law intuitively. The Fibonacci sequence represents how a population (e.g., rabbits) grows in a resource-unlimited environment. At a steady breeding speed, it takes much longer time to increase the population from 100 to 200 (need to increase by 100) than from 90 to 100 (only need to increase by 10). It also takes longer time than 200 to 300 because the population has grown bigger in the latter case. Therefore, it stays longer at smaller leading digits than the bigger ones.

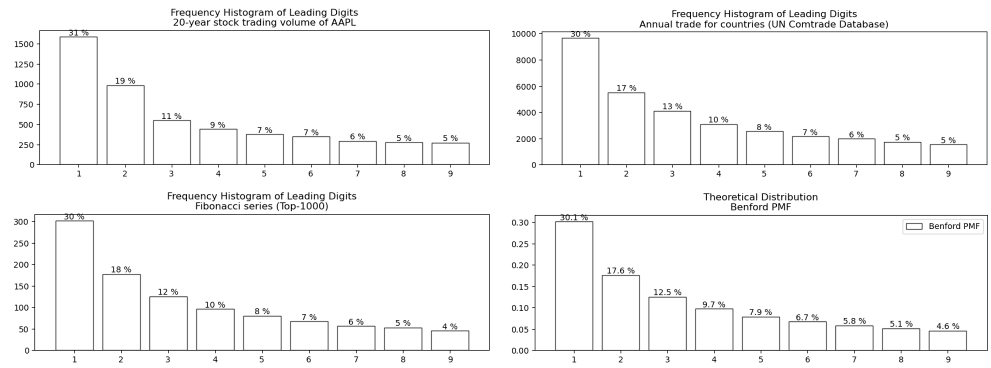


Figure 6: Use two real-life datasets and the Fibonacci series to verify the Benford law.

3. Functions for sampling distributions

The mc.samplings module provides functions for verifying sampling of common hypothesis test statistics(Table 5). Each function constructs the test statistic's random variable and compares it with the theoretical sampling distribution.

3.1. Student's t test

In t_stat(), we sample n samples from a normal distribution, the statistic will follow the student's distribution(Figure 7).

mc-tk function	Statistical Testing / Task	H_0 and population distribution	Test Statistic	Sampling distribution
mc.samplings. t_stat()	Student's t test	$H_0 : \mu = \mu_0$	$t = \frac{\overline{x}-\mu}{\frac{S}{\sqrt{n}}}$	$t(n-1)$
mc.samplings. chisq_gof_stat()	Pearson's Chi -squared GOF Test	$H_0 : PMF_1 = PMF_2$	$\chi^2 = \sum_{j=1}^{k} \frac{(f_j - np_j)^2}{np_j}$	$\chi^2(k-1)$
mc.samplings. anova_stat()	ANOVA	$H_0 : \mu_1 = \mu_2 = ... = \mu_k$	$F = \frac{MSTR}{MSE}$	$F(k-1, n-1)$
mc.samplings. kw_stat()	Kruskal-Wallis	H_0:All k populations have the same median	$H = [\frac{12}{n_T(n_T+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i}] - 3(n_T+1)$	$\chi^2(k-1)$
mc.samplings. fk_stat()	Fligner-Killeen Test	H_0:All k population variances are equal	$FK = \frac{\sum_{j=1}^{k} n_j(\overline{a_j}-\overline{a})^2}{s^2}$	$\chi^2(k-1)$
mc.samplings. bartlett_ stat()	Bartlett's Test	H_0:All k population variances are equal	$\chi^2 = \frac{(N-k)ln(S_p^2) - \sum_{i=1}^{k}(n_i-1)ln(S_i^2)}{1+\frac{1}{3(k-1)}(\sum_{i=1}^{k}(\frac{1}{n_i}) - \frac{1}{N-k})}$	$\chi^2(k-1)$
mc.samplings. sign_test_stat()	sign test	$H_0 : m = m_0$	N- N+	$b(n,p)$
mc.samplings. cochrange_q_stat()	Cochran Q Test	H_0:No difference between the k dichotomous populations	$T = \frac{(k-1)[k\sum_{j=1}^{k} X_{.j}^2 - (\sum_{j=1}^{k} X_{.j})^2]}{k\sum_{i=1}^{b} X_{i.} - \sum_{i=1}^{b} X_{i.}^2}$	$\chi^2(k-1)$
mc.samplings. hotelling_t2_stat()	Hotelling T^2 Test	$H_0 : \mu_1 = \mu 2$ μ_1 and μ_2 are vectors	$T^2 = n(\overline{x}-\mu)^T S^{-1}(\overline{x}-\mu)$	$T^2(k, n_k - 1)$
mc.samplings. clt()	Central Limit Theorem	$x_1, x_2, ..., x_n$ are n i.i.d. r.v.s.	$\overline{x} = \frac{x_1+x_2+...+x_n}{n}$	$N(\mu, \sigma^2)$

Table 5: Sampling distributions for commonly used hypothesis tests

$$\frac{\overline{x} - \mu}{\frac{S}{\sqrt{n}}} \sim t(n-1) \tag{8}$$

```
student(n=10,N=10000)
# n : sample size.
# N : how many MC experiments to run.
```

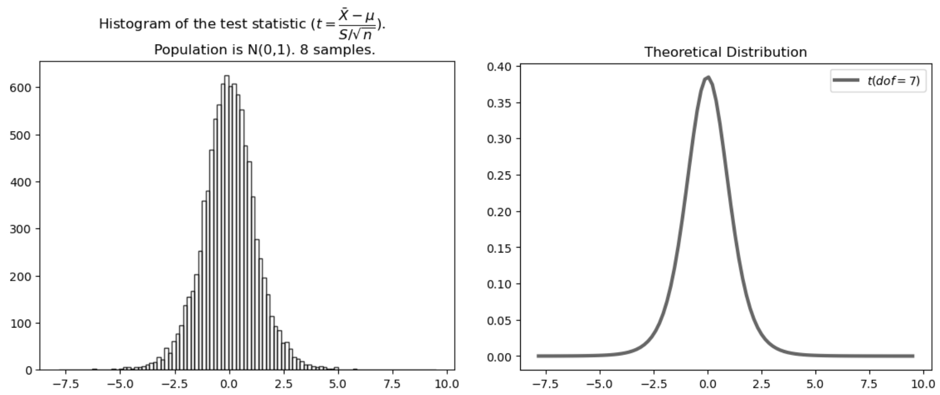


Figure 7: The observed frequency histogram and the corresponding theoretical t distribution (dof = 5).

3.2. Pearson's Chi-Square Goodness-of-Fit Test

Pearson's Chi-Square Goodness-of-Fit (GOF) test uses the following statistic.

$$\chi^2 = \sum_{j=1}^{k} \frac{(f_j - np_j)^2}{np_j} \sim \chi^2(k-1) \tag{9}$$

When n is large enough ($n \geq 50$), χ^2 will follow the $\chi^2(k-1)$ distribution. We can use chisq_gof_stat() to verify this sampling distribution intuitively. Because Pearson's chi-square GOF test is non-parametric, there is no restriction on the population distribution. chisq_gof_stat() provides two MC experiment settings. (1) The first is the Galton board (use the binominal population, Figure 8). (2) The second is the dice game (use the uniform PMF, Figure 9). In both cases, the statistic histogram from the MC experiment is very close to the theoretical $\chi^2(k-1)$ distribution.

```
chisq_gof_stat(dist='binom',K=8,sample_size=100,N=10000)
# dist : what kind of population dist to use.
#         By default, we use binom, i.e., the Galton board.
#    'binom'/'galton' - the population is binom.
#    'dice' - 6 * 1/6.
```

```
# K : classes in the PMF.
# N : how many MC experiments to run.
```

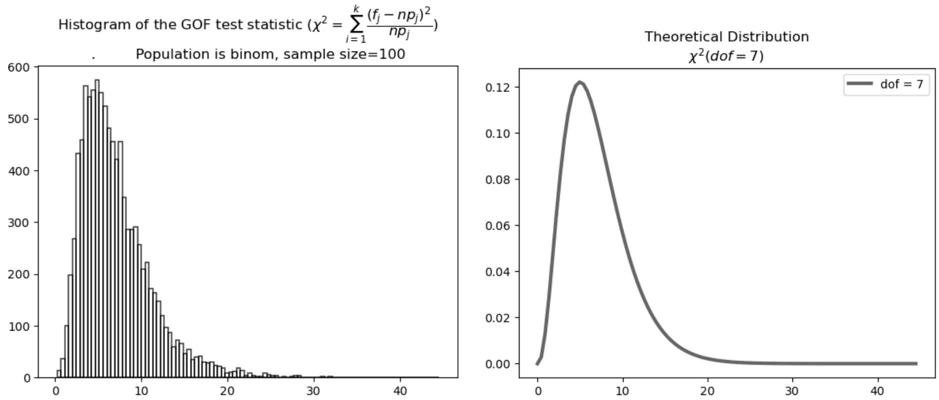


Figure 8: Use the Galton Board game to verify the statistic in Pearson's chi-square GOF test.

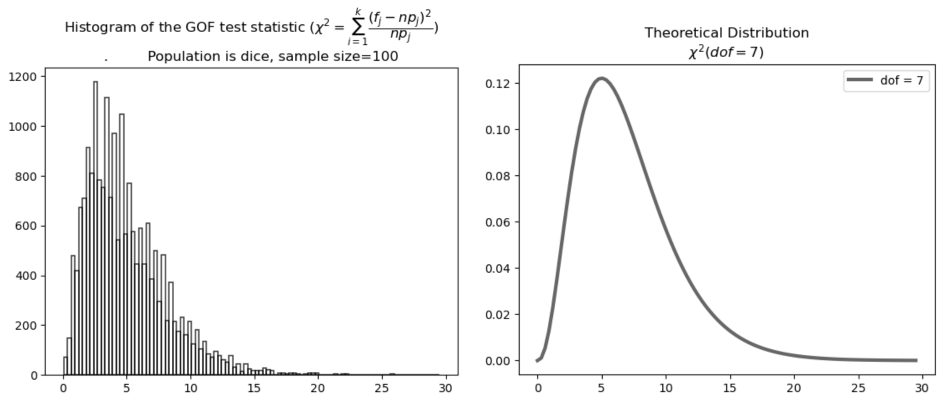


Figure 9: Use the dice game to verify the statistic in Pearson's chi-square GOF test.

3.3. ANOVA

ANOVA (analysis of variance) is a famous parametric mean test for multiple groups. Its null hypothesis H_0 is: $\mu_1 = \mu_2 = ... = \mu_k$. ANOVA constructs the test statistic by splitting the total variance into treatment (between-class difference, MSTR) and noise (within-class variance, MSE). When H_0 is true, the ratio of MSTR and MSE will follow the F distribution, i.e., $F = \frac{MSTR}{MSE} \sim F(k-1, n-1)$.

The anova_stat() will calculate the histogram of the F statistic drawn from a standard normal distribution and compare it to the theoretical F distribution (Figure 10).

```
anova_stat(K=10,n=10,N=10000)
# K : the number of classes/groups.
# n : the sample size in each class/group.
#     The total sample size is [K]*[n].
# N : how many MC experiments to run.
```

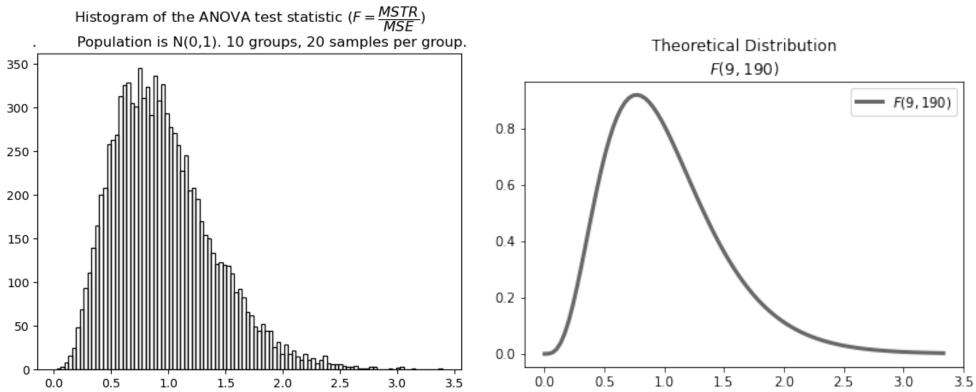


Figure 10: Use MC to verify the ANOVA test statistic follows the F distribution.

3.4. Kruskal-Wallis test

Kruskal-Wallis (K-W test) is the non-parametric counterpart for ANOVA. It can be applied to non-normal populations. Kruskal-Wallis is also an extension to the non-parametric Mann-Whitney or Wilcoxon rank sum test. The latter compares two groups, while Kruskal-Wallis can compare three or more.

The null hypothesis of the K-W test assumes that all populations have the same median. K-W is a rank-based test. Its H statistic is defined as:

$$H = [\frac{12}{n_T(n_T+1)}\sum_{i=1}^{k}\frac{R_i^2}{n_i}] - 3(n_T+1) \tag{10}$$

k is number of populations. n_i is the number of observations in group/sample I, $n_T = \sum_{i=1}^{k} n_i$ is the total number of observations in all samples. R_i is the sum of the ranks for sample i.

The kw_stat() function verifies that H follows the chi-square distribution (Figure 11).

```
kw_stat(dist='uniform',K=3,n=100,N=10000)
# dist : population assumption.
#        As the KW test is non-parametric,
#        the choice of dist doesn't matter.
#        By default, we use uniform.
```

```
# K : the number of groups/classes.
# n : the sample size in each group/class.
#     In this experiment, the sample size of each group
#     we used is equal, i.e., n1=n2=n3=...
# N : how many MC experiments to run.
```

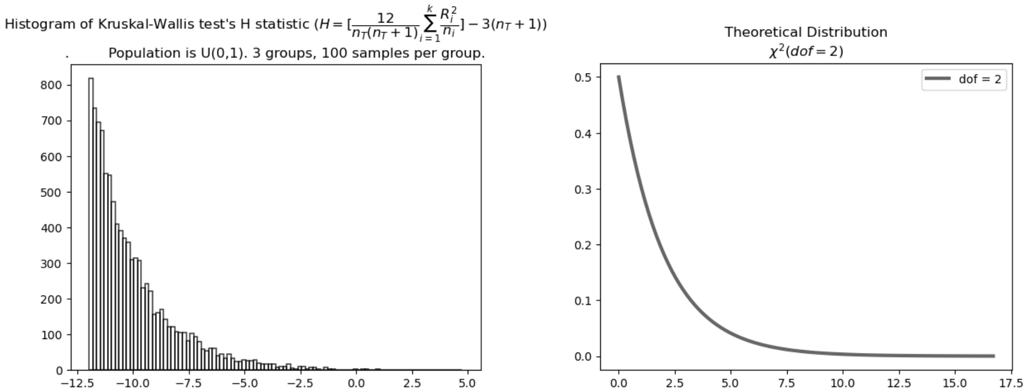


Figure 11: The Kruskal Wallis H statistic follows the chi-square distribution.

3.5. Fligner-Killeen Test

The Fligner-Killeen test is a non-parametric test for homogeneity of group variances based on ranks. It is a robust HOV (homogeneity of variances) test against departures from normality. It is more robust than the Levene test. The Fligner Killeen statistic is $FK = \frac{\sum_{j=1}^{k} n_j(\overline{a_j}-\overline{a})^2}{s^2}$, k is the number of groups, n_j is the size of the jth group, $\overline{a_j}$ is the group mean, $\overline{a}$ is the grand mean and s^2 is the grand variance. The fk_stat() function verifies the FK statistic follows the chi-square sampling distribution (Figure 12).

```
fk_stat(n=10,k=5,N=1000)
# n : the sample size in each group/class. In this
#     experiment, all group sizes are equal.
# K : the number of groups/classes.
# N : how many MC experiments to run.
```

3.6. Bartlett's Test

Bartlett's test is another HOV test. Bartlett's test is sensitive to departures from normality. Its test statistic is as follows.

$$\chi^2 = \frac{(N-k)ln(S_p^2) - \sum_{i=1}^{k}(n_i-1)ln(S_i^2)}{1 + \frac{1}{3(k-1)}(\sum_{i=1}^{k}(\frac{1}{n_i}) - \frac{1}{N-k})} \tag{11}$$

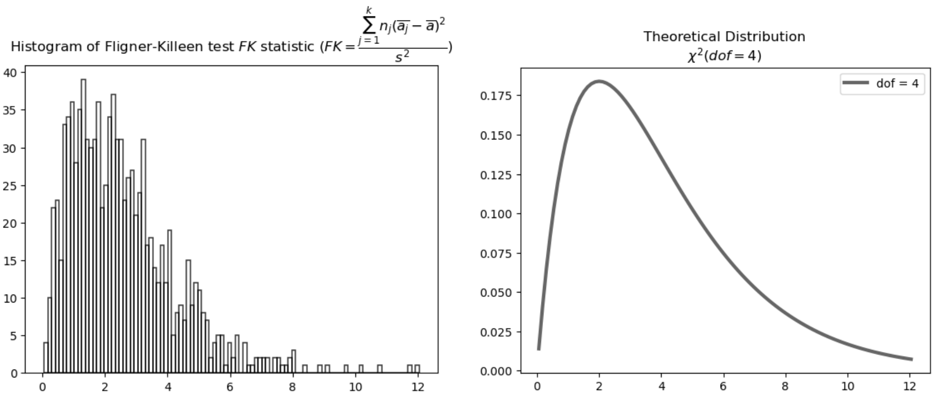


Figure 12: The observed histogram and theoretical distribution of the Fligner-Killeen statistic.

Where k is the group number. n_i and S_i^2 are the ith group size and variance. $N = \sum_{i=1}^{k} n_i$ is the grand total. $S_P^2 = \frac{1}{N-k} \sum_i (n_i - 1)$. S_i^2 is the pooled variance.

mc-tk provides the bartlett_stat() function to verify the test statistic follows a distribution (Figure 13).

```
bartlett_stat(k=5,n=10,N=1000)
# k : the number of groups/classes.
# n : the sample size in each group/class. In this
#     experiment, all group sizes are equal.
# N : how many MC experiments to run.
```

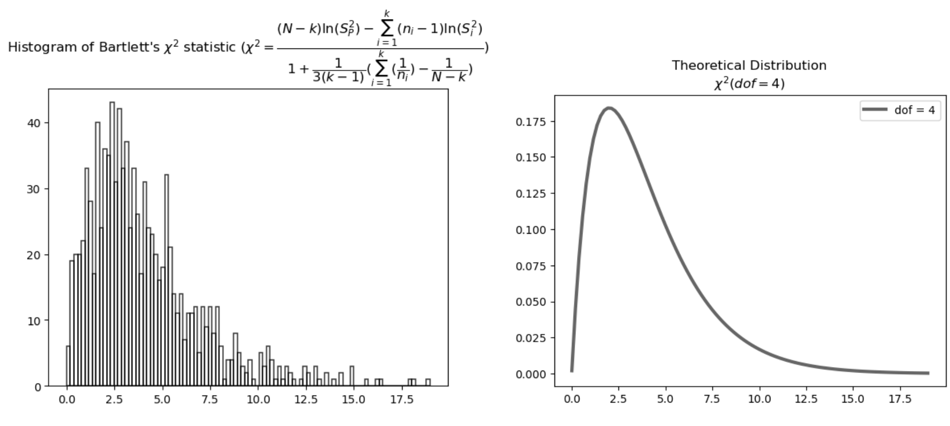


Figure 13: The observed histogram and theoretical distribution of Bartlett's test statistic.

3.7. Sign Test

The sign test is a widely used non-parametric median test. The sign test first subtracts each sample by the grand median, i.e., $X_i - m_0$,$i = 1, 2, ..., n$. Then, it counts positive and negative numbers denoted as N+ and N-. If the null hypothesis ($H_0 : m = m_0$) is true, N+ and N- will both follow binomial distributions:

$$N- \sim b(n, \frac{1}{2}) \qquad N+ \sim b(n, \frac{1}{2}) \tag{12}$$

To verify this sampling distribution, we can use the sign_test_stat() function. This function uses the exponential distribution for demonstration. By solving $\int_m^\infty \theta e^{-\theta x} dx = 1/2$, we can get the theoretical median is $m = \theta ln^{(2)}$. The sign_test_stat() function draws samples from the exponential population and compares each sample with the theoretical median to get the N+ and N- statistic values. The result is shown in Figure 14.

```
sign_test_stat(dist='expon',n=100,N=10000)
# dist : population assumption.
#        As the sign test is non-parametric,
#        the choice of dist doesn't matter.
#        By default, we use exponential.
# n : sample size.
# N : how many MC experiments to run.
```

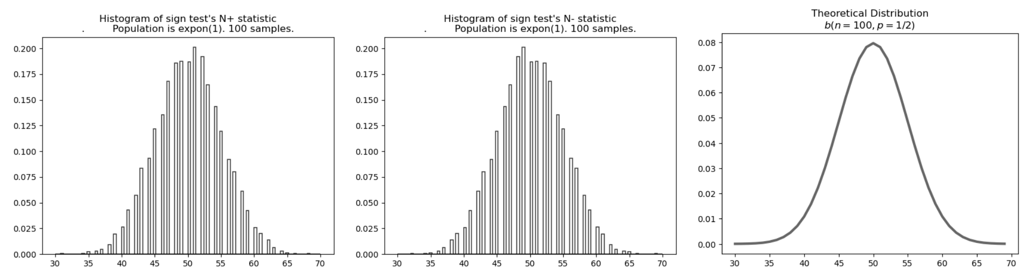


Figure 14: Use the exponential distribution to verify the N+ and N- follow the binomial distributions.

3.8. Cochrane-Q Tes

The Cochran's Q test is a non-parametric test to check whether three or more dichotomous / Boolean groups have the same True or False proportions. This test uses the T statistic.

$$T = \frac{(k-1)[k\sum_{j=1}^{k} X_{.j}^2 - (\sum_{j=1}^{k} X_{.j})^2]}{k\sum_{i=1}^{b} X_{i.} - \sum_{i=1}^{b} X_{i.}^2} \tag{13}$$

k: The number of treatments / groups
$X_{.j}$: The column total for the treatment
b: The number of rows (i.e., samples per class)
$X_{i.}$: The total for the i^{th} row
N: The grand total

The cochrane_q_stat() function verifies T follows the $\chi^2(k-1)$ distribution. Because Cochrane's Q requires Boolean data, cochrane_q_stat() uses the Bernoulli distribution to generate samples (Figure 15).

```
cochrane_q_stat(p=0.5,K=3,n=100,N=10000)
# p : we draw from a Bernoulli population with p.
#     p is the "success/pass" probability.
# K : the number of groups/classes.
# n : the sample size in each group/class. In this
#     experiment, all group sizes are equal,
#     as Cochrane-Q is paired / dependent.
# N : how many MC experiments to run.
```

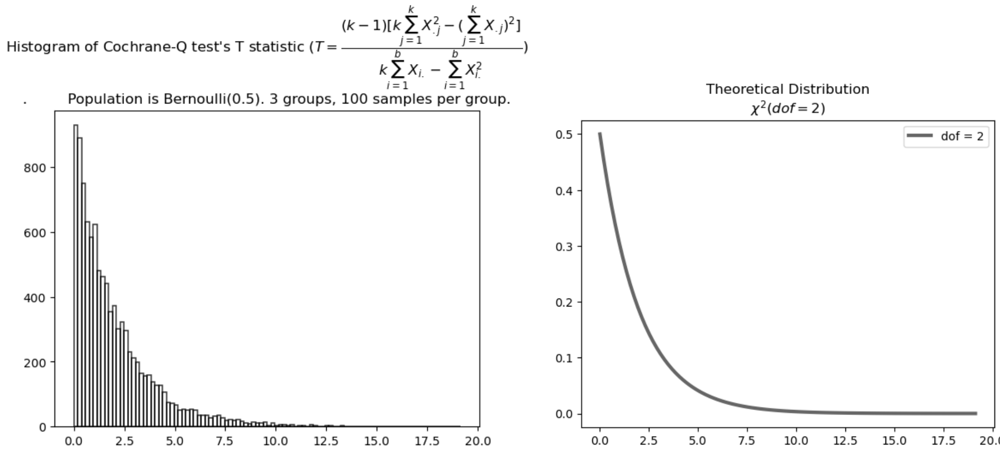


Figure 15: The T statistic of Cochran's Q test and its theoretical chi-square distribution.

3.9. Hotelling's T^2 Test

The Hotelling's T^2 test compares the mean of two multivariate populations. Suppose we have two groups of samples from $N(\mu_1, \sum)$ and $N(\mu_2, \sum)$. They share the same covariance matrix $\sum$. The null hypothesis is $H_0 : \mu_1 = \mu_2$ and the test statistic is:

$$T^2 = n(\overline{x} - \mu)^T S^{-1}(\overline{x} - \mu) \quad (14)$$

$S = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(x_i - \overline{x})^T$ is the grand covariance matrix.

If the dimensionality k=1, Hotelling's T^2 degenerates into the t distribution. When $K \geq 2$, it is a multivariate generalization of the t distribution. The hotelling_t2_stat() verifies the T^2 sampling distribution (Figure 16).

```
hotelling_t2_stat(n=50,k=2,N=1000)
# n : the sample size in each class.
# k : data dimension.
# N : how many MC experiments to run.
```

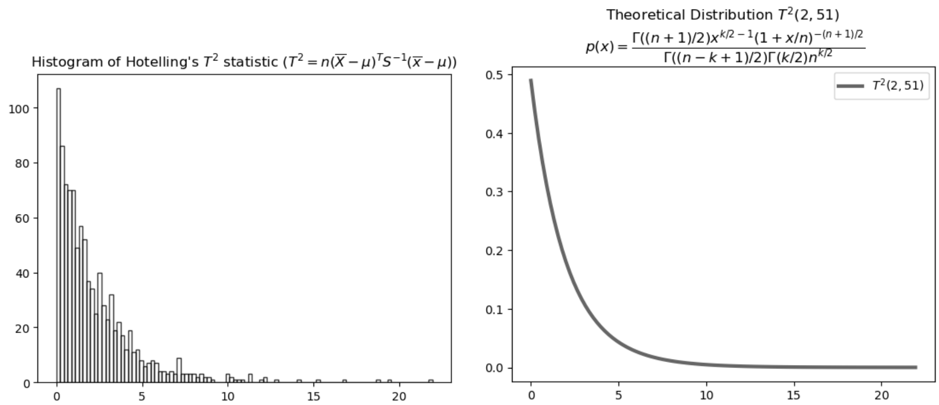
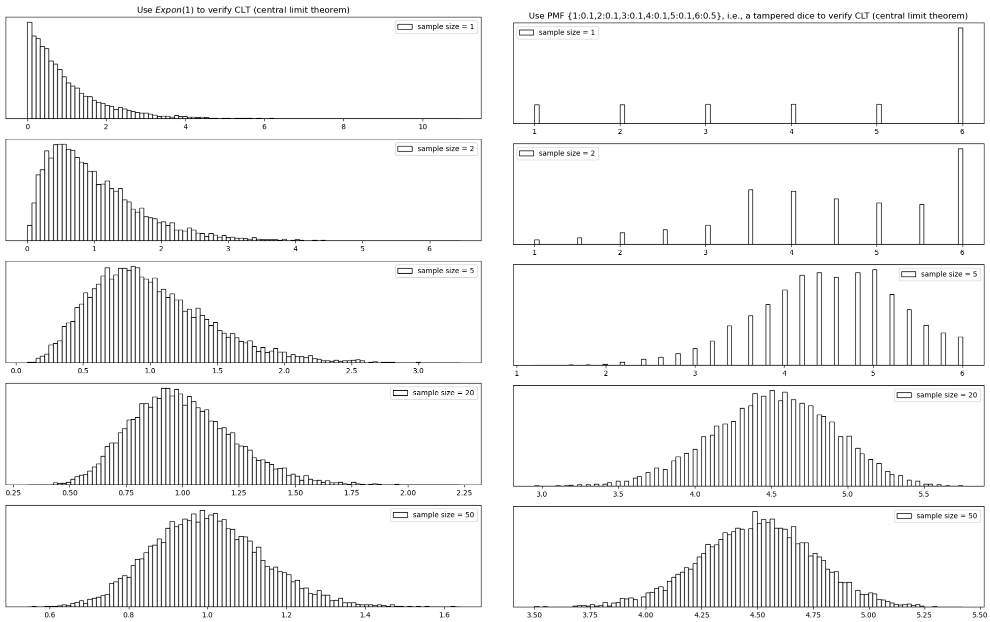


Figure 16: The T^2 statistic of Hotelling's test.

3.10. Central Limit Theorem

The Central Limit Theorem (CLT) is a fundamental theory in statistics. It explains why so many physical phenomena follow the normal distribution. No matter the underlying distribution, the sum/average of enough i.i.d. r.v.s. will approach the normal distribution. The clt() function provides these underlying distributions to test CLT. (1) uniform, (2) exponential, see Figure 17a, (3) Poisson, (4) coin, i.e., Bernoulli with $p = 0.5$, (5) tampered coin, e.g., Bernoulli with p = 0.8, (6) dice, i.e., $p(k) = 1/6$, (7) tampered dice, e.g., $p(k = 6) > 1/6$. See Figure 17b.

```
clt(dist='bernoulli',sample_size=[1,2,5,20],
  N=10000,display=True)
# dist : base/undeyling/atom distribution.
#     'uniform' - a uniform distribution U(-1,1) is used.
#     'expon' - an exponential distribution Expon(1)
#             is used.
#     'poisson' - poisson distribution PI(1) is used.
#     'coin' / 'bernoulli' - {0:0.5,1:0.5}.
#     'tampered_coin' - {0:0.2,1:0.8}.
#     'dice' - {1:1/6,2:1/6,3:1/6,4:1/6,5:1/6,6:1/6}
#     'tampered_dice' - {1:0.1,2:0.1,3:0.1,4:0.1,5:0.1,6:0.5}
#                       6 is more likely.
#     None - use 0-1 distribution {0:0.5,1:0.5} by default.
# sample_size : the number of samples to be averaged over/summed up.
```



(a) (b)

Figure 17: Use the exponential (a) and a tampered dice (e.g., six has a higher probability) (b) to verify CLT.

```
#                 It can be an array/list , and users can observe how
#                 the histogram changes as the sample size varies.
# N : how many MC experiments to run.
```

References

[1] C.-O. Hwang, Y. Kim, C. Im, S. Lee, Buffon's needle algorithm to estimate π, Applied Mathematics 08 (2017) 275–279. `doi:10.4236/am.2017.83022`.

[2] A. Gál, P. B. Miltersen, The cell probe complexity of succinct data structures, Theoretical Computer Science 379 (3) (2007) 405–417, automata, Languages and Programming. `doi:https://doi.org/10.1016/j.tcs.2007.02.047`.
URL `https://www.sciencedirect.com/science/article/pii/S030439750700151X`

[3] M. Warshauer, E. Curtin, The locker puzzle, The Mathematical Intelligencer 28 (2006) 28–31. `doi:10.1007/BF02986999`.

[4] J. Kanwal, K. Smith, J. Culbertson, S. Kirby, Zipf's law of abbreviation and the principle of least effort: Language users optimise a miniature lexicon for efficient communication, Cognition 165 (2017) 45–52. doi:https://doi.org/10.1016/j.cognition.2017.05.001.
URL https://www.sciencedirect.com/science/article/pii/S0010027717301166

[5] A. Cerioli, L. Barabesi, A. Cerasa, M. Menegatti, D. Perrotta, Newcomb-benford law and the detection of frauds in international trade, Proceedings of the National Academy of Sciences 116 (1) (2019) 106–115. arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.1806617115, doi:10.1073/pnas.1806617115.
URL https://www.pnas.org/doi/abs/10.1073/pnas.1806617115