基礎数理学統計レポート

１．２．

シミュレーション方法

* 使用言語：Python（NumPy, SciPy）
* 信頼区間：95%信頼区間（t分布を使用）
* 分布：一様分布 U(0,1)、指数分布（平均1）
* サンプルサイズ n：5, 10, 30, 50, 100, 200
* 繰り返し回数：5000回
* 評価指標：カバレッジ率（真の平均が区間内に含まれた回数 / 5000）

結果は次のとおりである．

|  |  |  |
| --- | --- | --- |
| **サンプルサイズ n** | **Uniform(0,1)** | **Exponential(1)** |
| 5 | 0.9350 | 0.8830 |
| 10 | 0.9474 | 0.8990 |
| 30 | 0.9444 | 0.9346 |
| 50 | 0.9448 | 0.9300 |
| 100 | 0.9476 | 0.9460 |
| 200 | 0.9476 | 0.9500 |

シミュレーション方法

• 使用言語：R（base＋statsパッケージ）  
• 信頼区間：95%信頼区間（t分布を使用）  
• 分布：一様分布 U(0,1)、指数分布（平均1）  
• サンプルサイズ n：5, 10, 30, 50, 100, 200  
• 繰り返し回数：5000回  
• 評価指標：カバレッジ率（真の平均が区間内に含まれた回数／5000）

結果は次のとおりである．

|  |  |  |
| --- | --- | --- |
| **サンプルサイズ n** | **Uniform(0,1)** | **Exponential(1)** |
| 5 | 0.9344 | 0.8926 |
| 10 | 0.9440 | 0.9056 |
| 30 | 0.9532 | 0.9302 |
| 50 | 0.9454 | 0.9348 |
| 100 | 0.9462 | 0.9472 |
| 200 | 0.9498 | 0.9474 |

シミュレーション方法（MATLAB版）  
• 使用言語：MATLAB（Statistics and Machine Learning Toolbox）  
• 信頼区間：95%信頼区間（t 分布を使用）  
• 分布：一様分布 U(0,1)、指数分布（平均1）  
• サンプルサイズ n：5, 10, 30, 50, 100, 200  
• 繰り返し回数：5000回  
• 評価指標：カバレッジ率（真の平均が区間内に含まれた回数／5000回）

|  |  |  |
| --- | --- | --- |
| **サンプルサイズ nnn** | **Uniform(0,1)** | **Exponential(1)** |
| 5 | 0.9390 | 0.8808 |
| 10 | 0.9502 | 0.9002 |
| 30 | 0.9506 | 0.9320 |
| 50 | 0.9462 | 0.9324 |
| 100 | 0.9560 | 0.9444 |
| 200 | 0.9494 | 0.9456 |

# pip install numpy scipy

import numpy as np

from scipy import stats

def simulate\_coverage(dist, params, ns, reps=5000, alpha=0.05):

"""

dist: 'uniform' または 'exponential'

params: {'low': a, 'high': b} もしくは {'scale': λ の逆数}

ns: サンプルサイズリスト

reps: 繰り返し回数

alpha: 有意水準

"""

results = []

if dist == 'uniform':

a, b = params['low'], params['high']

true\_mean = (a + b) / 2

sampler = lambda n: np.random.uniform(a, b, size=n)

else: # exponential

scale = params['scale']

true\_mean = scale

sampler = lambda n: np.random.exponential(scale, size=n)

for n in ns:

count = 0

for \_ in range(reps):

x = sampler(n)

m = x.mean()

s = x.std(ddof=1)

ci\_low, ci\_high = stats.t.interval(1 - alpha, df=n-1, loc=m, scale=s/np.sqrt(n))

if ci\_low <= true\_mean <= ci\_high:

count += 1

coverage = count / reps

results.append((n, coverage))

return results

if \_\_name\_\_ == "\_\_main\_\_":

ns = [5, 10, 30, 50, 100, 200]

uni = simulate\_coverage('uniform', {'low': 0, 'high': 1}, ns)

exp = simulate\_coverage('exponential', {'scale': 1}, ns)

print("Uniform(0,1) の 95% 信頼区間カバレッジ率")

for n, cov in uni:

print(f" n={n:<3} → coverage={cov:.4f}")

print("\nExponential(mean=1) の 95% 信頼区間カバレッジ率")

for n, cov in exp:

print(f" n={n:<3} → coverage={cov:.4f}")

# シミュレーション用関数定義（Python）

simulate\_coverage <- function(dist, params, ns, reps = 5000, alpha = 0.05) {

coverages <- numeric(length(ns))

for (i in seq\_along(ns)) {

n <- ns[i]

count <- 0

for (j in seq\_len(reps)) {

x <- if (dist == "uniform") {

runif(n, min = params$low, max = params$high)

} else {

rexp(n, rate = 1 / params$scale)

}

ci <- t.test(x, conf.level = 1 - alpha)$conf.int

true\_mean <- if (dist == "uniform") {

(params$low + params$high) / 2

} else {

params$scale

}

if (ci[1] <= true\_mean && true\_mean <= ci[2]) {

count <- count + 1

}

}

coverages[i] <- count / reps

}

data.frame(n = ns, coverage = coverages)

}

# サンプルサイズ

ns <- c(5, 10, 30, 50, 100, 200)

# 一様分布と指数分布のシミュレーション実行

uni\_df <- simulate\_coverage(

dist = "uniform",

params = list(low = 0, high = 1, scale = NULL),

ns = ns

)

exp\_df <- simulate\_coverage(

dist = "exponential",

params = list(low = NULL, high = NULL, scale = 1),

ns = ns

)

# 結果表示

print("Uniform(0,1) の 95% 信頼区間カバレッジ率")

print(uni\_df)

print("Exponential(mean=1) の 95% 信頼区間カバレッジ率")

print(exp\_df)% シミュレーション方法（MATLAB版）

% 使用言語：MATLAB（Statistics and Machine Learning Toolbox）

% 信頼区間：95%信頼区間（t分布を使用）

% 分布：一様分布 U(0,1)、指数分布（平均1）

% サンプルサイズ n：5, 10, 30, 50, 100, 200

% 繰り返し回数：5000回

% 評価指標：カバレッジ率（真の平均が区間内に含まれた回数／5000回）

ns = [5, 10, 30, 50, 100, 200];

reps = 5000;

alpha = 0.05;

cover\_uniform = zeros(size(ns));

cover\_exp = zeros(size(ns));

for i = 1:numel(ns)

n = ns(i);

tcrit = tinv(1 - alpha/2, n - 1);

count\_u = 0;

count\_e = 0;

for j = 1:reps

% --- 一様分布 U(0,1) ---

x = rand(n, 1);

m = mean(x);

s = std(x); % 標本標準偏差 (n-1 で除算)

se = s / sqrt(n);

ci = [m - tcrit\*se, m + tcrit\*se];

if ci(1) <= 0.5 && 0.5 <= ci(2)

count\_u = count\_u + 1;

end

% --- 指数分布 (mean = 1) ---

y = exprnd(1, n, 1);

m = mean(y);

s = std(y);

se = s / sqrt(n);

ci = [m - tcrit\*se, m + tcrit\*se];

if ci(1) <= 1 && 1 <= ci(2)

count\_e = count\_e + 1;

end

end

cover\_uniform(i) = count\_u / reps;

cover\_exp(i) = count\_e / reps;

end

% 結果表示

disp('Uniform(0,1) の 95% 信頼区間カバレッジ率');

for i = 1:numel(ns)

fprintf(' n=%3d → coverage=%.4f\n', ns(i), cover\_uniform(i));

end

disp('Exponential(mean=1) の 95% 信頼区間カバレッジ率');

for i = 1:numel(ns)

fprintf(' n=%3d → coverage=%.4f\n', ns(i), cover\_exp(i));

end