

수치해석 과제 04 - 난수 발생기

2017030500 홍승표

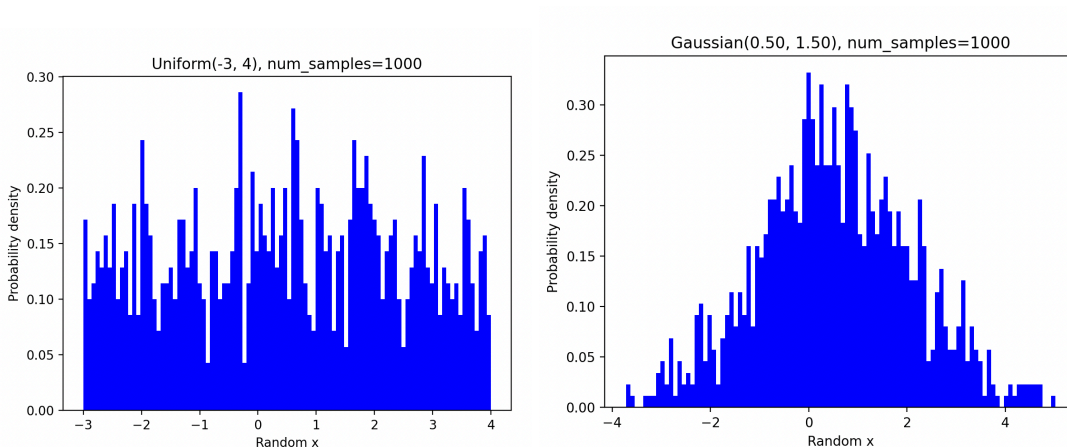
요구사항

1. 1,000개의 uniform, gaussian 난수 발생 후 100개 interval 로 히스토그램 생성
2. 요구사항 1의 샘플 개수를 100, 10000, 100000으로 변화
3. 요구사항 2의 변화에 따른 히스토그램 형상 변화 논의

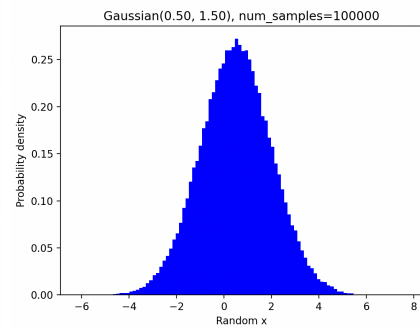
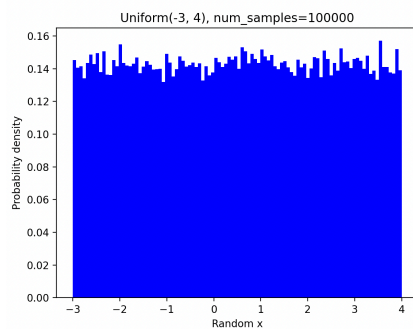
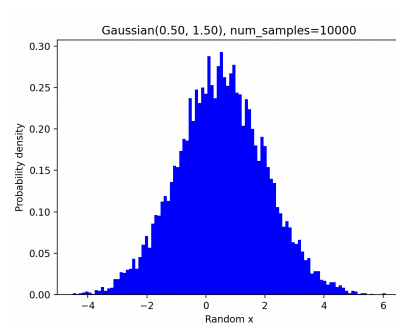
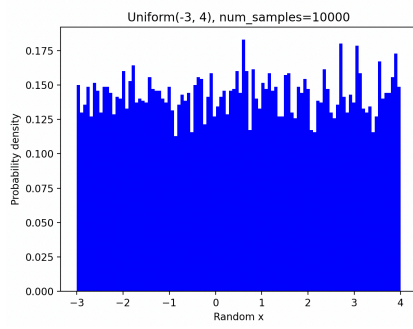
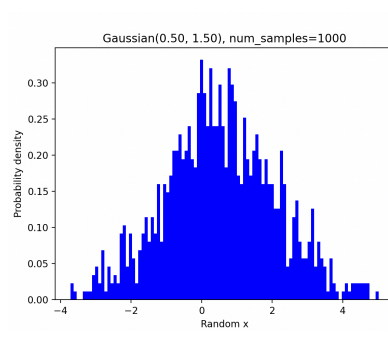
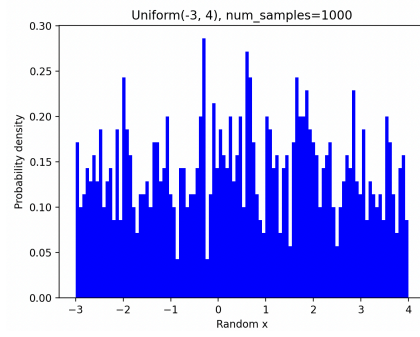
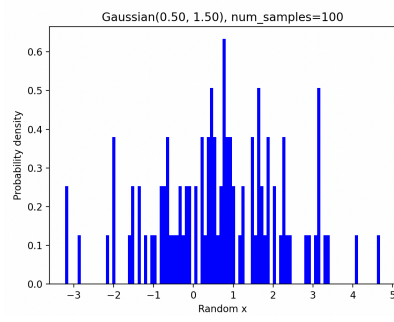
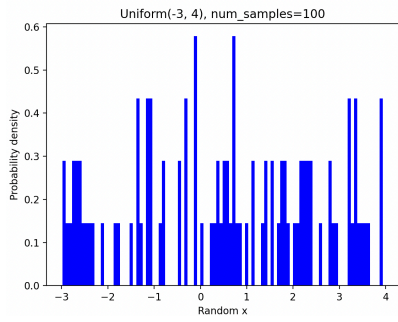
구현 방법

- 임의의 PDF를 표현 가능한 함수 포인터 타입 정의 (sampler_t)
- sampler_t, 샘플 개수를 받고, $scale * sampler() + shift$ 꼴의 선형 변환 함수 zzamtong 정의
 - uniform: $scale=b-a$, $shift=a$
 - gaussian: $scale=s$, $shift=m$
- Python 으로 시각화하기 위해 STDOUT 출력
 - 첫 줄은 히스토그램 제목 출력
 - 다음 줄부터는 한 줄에 한개의 난수 float 값을 출력
- C 난수 생성 코드는 Appendix A와 같음
- Python 히스토그램 시각화 코드는 Appendix B와 같음

시각화 결과 및 논의



- 1,000개 샘플에 대해, 각 PDF의 개형과 대략적으로 비슷하나 오차가 보임
- 샘플 개수를 100, 1000, 10000, 100000 과 같이 늘리면 아래와 같이 각 PDF의 알려진 개형에 갈수록 근접해 가는 모습을 볼 수 있음



Appendix

A. C 난수 생성 코드 (main.c)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#include "nr.h"

typedef float (*sampler_t)(long *);

void zzamtong(int num_samples, sampler_t sampler, long *idum, float scale, float shift) {
    for (int i = 0; i < num_samples; i++) {
        printf("%f\n", sampler(idum) * scale + shift);
    }
}

void run_uniform(int num_samples, long *idum) {
    int a = -3;
    int b = 4;
    printf("Uniform(%d, %d), num_samples=%d\n", a, b, num_samples);
    zzamtong(num_samples, ran1, idum, b - a, a);
}

void run_gaussian(int num_samples, long *idum) {
    float m = 0.5;
    float s = 1.5;
    printf("Gaussian(%.2f, %.2f), num_samples=%d\n", m, s, num_samples);
    zzamtong(num_samples, gasdev, idum, s, m);
}

int main(int argc, char *argv[]) {
    if (argc < 3) {
        printf("Usage: %s <num_samples> <PDF: 'u' or 'g'>\n", argv[0]);
        return 1;
    }
    int num_samples = atoi(argv[1]);
    long idum = time(NULL);
    char pdf = argv[2][0];
    if (pdf == 'u') {
        run_uniform(num_samples, &idum);
    } else if (pdf == 'g') {
        run_gaussian(num_samples, &idum);
    } else {
        printf("Invalid PDF: Expected 'u' or 'g', got '%c'\n", pdf);
        return 1;
    }
    return 0;
}
```

B. Python 히스토그램 시각화 코드 (show.py)

```
import sys
import matplotlib.pyplot as plt

v = [l.strip() for l in sys.stdin]
plt.title(v[0])
plt.hist(
    [float(x) for i, x in enumerate(v) if i > 0], bins=100, density=True, color="b"
)
plt.xlabel("Random x")
plt.ylabel("Probability density")
plt.show()
```