

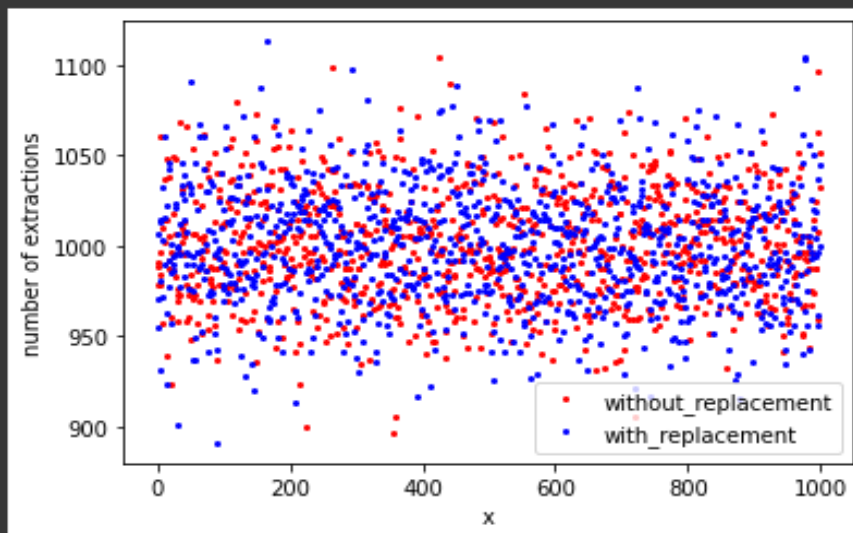
[과제] Reservoir Sampling / DGIM Algorithm

학번 : 20181612

이름 : 박병규

Reservoir Sampling 결과

```
plt.plot(x,res_rep, 'bo', markersize = 2, label="with_replacement")
plt.legend()
plt.show()
```



비복원추출일 때
최솟값: 897, 최댓값: 1104, 평균값: 1000.0

복원추출일 때
최솟값: 891, 최댓값: 1113, 평균값: 1000.0

0부터 999까지 100개의 숫자를 뽑는 시행을 복원, 비복원으로 10000번 실행해본 결과

비복원추출일 때

가장 적게 뽑힌 횟수: 897 , 가장 많이 뽑힌 횟수: 1104

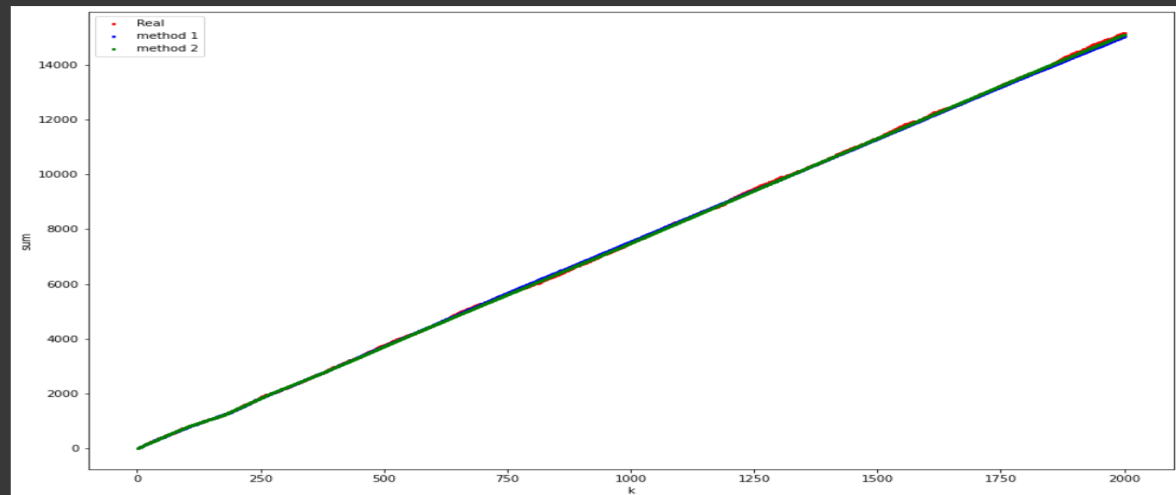
복원추출일 때

가장 적게 뽑힌 횟수: 891 , 가장 많이 뽑힌 횟수: 1113

Boxplot에서 IQR()값을 보게되면 중앙값 근처에 균등하게 뽑혔음을 알 수 있다.

DGIM Algorithm 결과

```
plt.figure(figsize=(10,10))
x = [i for i in range(1,2001)]
plt.plot(x, sum_real, 'ro', label = "Real", markersize = 2)
plt.plot(x, sum1, 'bo', label = "method 1", markersize = 2)
plt.plot(x, sum2, 'go', label = "method 2", markersize = 2)
plt.xlabel('k')
plt.ylabel('sum')
plt.legend()
plt.show()
```



오차범위 1%일 때
방법 1의 정확도는 0.8255
방법 2의 정확도는 0.886

오차범위 2%일 때
방법 1의 정확도는 0.971
방법 2의 정확도는 0.9855

오차범위 3%일 때
방법 1의 정확도는 0.9875
방법 2의 정확도는 0.991



"""

오차율을 계산해 보자

"""

```
error1, error2 = 0, 0
for i in range(2000):
    error1 += abs(sum_real[i]-sum1[i])/sum_real[i]*100
    error2 += abs(sum_real[i]-sum2[i])/sum_real[i]*100
```

```
print("방법 1의 오차율의 평균 ", error1/2000)
print("방법 2의 오차율의 평균 ", error2/2000)
```

방법 1의 오차율의 평균 0.6376546098373639
방법 2의 오차율의 평균 0.4896257993129154

1%, 2%, 3% 의 오차범위를 인정하고 정확도를 계산 해본 결과

오차범위 1%일 때

방법 1의 정확도는 0.8255

방법 2의 정확도는 0.886

오차범위 2%일 때

방법 1의 정확도는 0.971

방법 2의 정확도는 0.9855

오차범위 3%일 때

방법 1의 정확도는 0.9875

방법 2의 정확도는 0.991

위의 결과를 바탕으로 방법2가 더 정확한 알고리즘임을 확인 할 수 있었다