

1121_SIMULATION AND STATISTICAL COMPUTING

#HW6

M/M/1 Simulation

設計模擬 M/M/1 演算法

1. 指數隨機變數產生器：

```
#指數隨機變數產生器
def generate_EXP(lambda_param):
    U = np.random.uniform(0, 1)
    X = -1/lambda_param * np.log(U)
    return X
```

透過過去作業所完成的隨機變數產生器，來模擬顧客到達時間與服務時間。

2. 產生隨機顧客到達時間：

```
# 產生到達時間
for i in range(num_customers):
    tp = generate_EXP(1/arrival_rate)
    arrival_times.append(arrival_times[-1] + tp)
```

由於顧客到達時間須滿足 Poisson process，因次我透過指數隨機變數產生器產生間格區間，再加上上一個顧客的到達時間，以符合 Poisson process。

3. 產生隨機顧客到達時間：

```
# 產生隨機服務時間
for i in range(num_customers):
    service_times.append(generate_EXP(1/service_rate))
print(service_times)
```

4. 服務顧客模擬：

```
# 服務顧客
for i in range(1, num_customers):
    starting_queue.append(max(arrival_times[i], departure_times[i-1]))
    departure_times.append(starting_queue[i] + service_times[i])
    working += service_times[i]
```

根據上一個顧客的結束時間與此顧客到達時間，來決定此顧客的開始服務時間並計算其結束服務時間。

每輪模擬所顯示的模擬結果：

Customer	Arrival Time	Service Start Time	Service End Time	Wait Time
1	0.0000	0.0000	15.8331	0.0000
2	5.9892	15.8331	37.7053	9.8439
3	30.2038	37.7053	46.9827	7.5015
4	31.4007	46.9827	57.1639	15.5821
5	40.7385	57.1639	62.1648	16.4253
6	52.4059	62.1648	62.1712	9.7589
7	55.7131	62.1712	69.3259	6.4581
8	58.8145	69.3259	71.4454	10.5114
9	60.3015	71.4454	77.6679	11.1440
10	65.1826	77.6679	78.6628	12.4854

Average Wait Time: 9.9711 minutes

Server Utilization: 0.7987

圖(一) 單位時間顧客到達數量：6 單位時間顧客服務率：8 顧客數量：10 的模擬結果

根據上圖的其中一次模擬結果可得，平均的等待時間為 9.9711 分鐘，且利用率約為 0.79%。預期的利用率約為 75%(6/8)，和模擬結果存在一些落差，這可能是隨機性所帶來的因素。

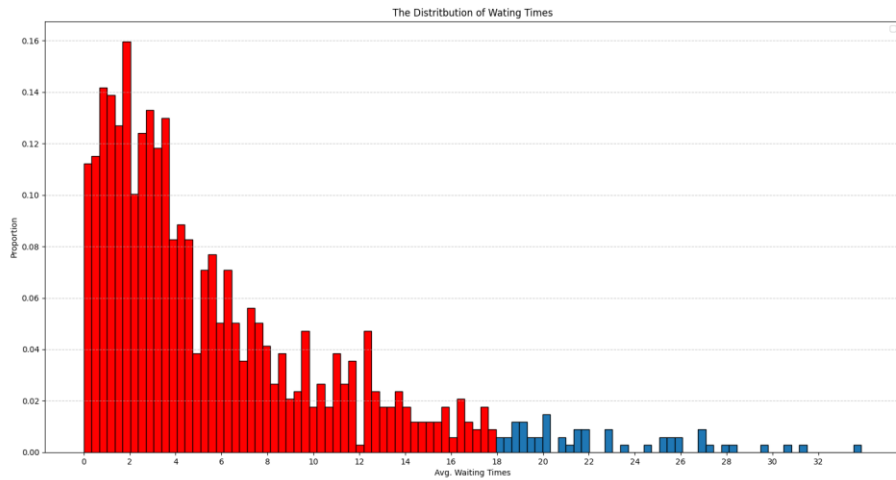
利用率是否可能達成 100%？

至於利用率我認為可能達成 100%，若 $\lambda = \mu$ ，平均到達率等於平均服務率。表示顧客到達速度和系統服務速度相等，可讓系統處於滿載。然而在實際情況中，利用率很難真正達成 100%，原因可能為：服務時間存在變化也存在隨機性，因此達成 100%在實務上相對困難。

為什麼等待時間分布圖不能使用間隔區間？

由於等待時間為一個連續型的變數型態，因此，若使用較大的間隔區間，較無法反映真實的分布情況，因此在接下來的模擬分布圖中，我也會將區間間隔盡量的縮小，盡可能模擬顯示連續型態的等待時間結果。

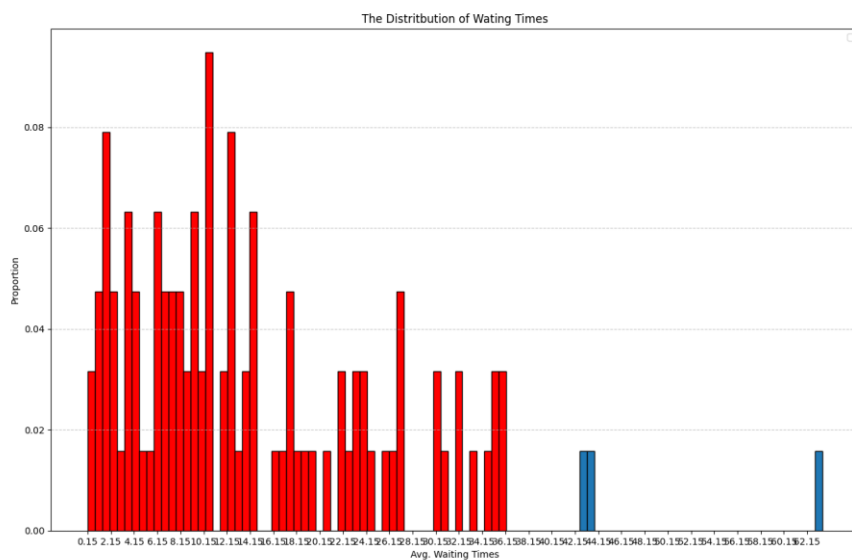
我將會測試多次的 M/M/1 模擬結果，找出平均的等待時間。



圖(二) 模擬 1000 次後每輪平均等待時間分布圖(紅色範圍為介於兩個標準差內的信賴區間)

透過 1000 次模擬，計算在兩個標準差範圍內的等待時間模擬數量，可計算得約為 95%，接近常態分佈的信賴區間。

0.955%



圖(三) 模擬 100 次後每輪平均等待時間分布圖(紅色範圍為介於兩個標準差內的信賴區間)

由模擬結果圖可觀察，若模擬次數較少，會造成分布圖較為零散，相對於多次的模擬，較難看出分布結果。