

通信システム実験2 待ち行列シミュレーション

信州大学工学部 電子情報システム工学科

実験日: 2023/11/01

実験場所: W2 棟 101 室

実験者: 21T2166D 渡辺 大樹

共同実験者: 20T2062A 斎藤 創真

21T2004H 朝日 純菜

21T2033A 岡村 椋平

21T2091J 徳竹 稜平

21T2099D 中田 奏音

1 実験内容

本実験ではネットワークにおけるパケット通信のレスポンスなどに用いられる待ち行列理論をバスの運行シミュレーションを通してその理論の体験、理解を進める目的で行う。

実験1ではベルヌーイ過程を、実験2では指数分布をシミュレーション、理解し、これらを用いて実験3でバスの運行シミュレーションを行う。

2 レポート課題

以下にレポート課題の解答を示す。

レポート課題 1.1

確率変数 Z の分布関数 $F_Z(x)$ を密度関数 $f_Z(x)$ で表すと

$$F_Z(x) = \int_{-\infty}^{\infty} f_Z(x) dx \quad (1)$$

となる。

レポート課題 1.2

到着確率の期待値が $\frac{\delta}{p}$ となることを示す。

タイムスロットの間隔を δ , 各スロットでの到着確率を p としたときの到着間隔の期待値を求める。到着間隔の確率変数を Z , 確率を $P_Z(n)$ とすると $n = 1, 2, 3, \dots$ に対して確率 $P_Z(n)$ は

$$\begin{aligned} P_Z(1) &= p \\ P_Z(2) &= p(1-p) \\ P_Z(3) &= p(1-p)^2 \end{aligned} \quad (2)$$

となる。期待値の式は確率変数とその時の確率を掛け合わせ、

$$E[Z] = \sum_{n=1}^{\infty} n \cdot \delta \cdot P_Z(n) \quad (3)$$

となる。

この式に先ほどの確率 $P_Z(n)$ の一般化した式を代入することで

$$E[Z] = \delta p \sum_{n=1}^{\infty} n(1-p)^{n-1} \quad (4)$$

という式が得られる。

ここでシグマの中身を部分和として計算していく。

$$S_n = \sum_{k=1}^n k(1-p)^{k-1} \quad (5)$$

とする。 S_n に $(1-p)$ を掛け算すると

$$(1-p)S_n = \sum_{k=1}^n k(1-p)^k \quad (6)$$

となる。(5) から (6) を引き算すると

$$pS_n = \sum_{k=1}^{n+1} (1-p)^{k-1} \quad (7)$$

という式が得られる。この式は初項 1 公比 $(1-p)$ の等比数列の和となるため公式を用いて計算すると

$$pS_n = \frac{1 - (1-p)^{n+1}}{p} \quad (8)$$

となる。すなわち S_n は

$$S_n = \frac{1 - (1-p)^{n+1}}{p^2} \quad (9)$$

となる。

S_n は部分和のため n について極限をとることで

$$\sum_{n=1}^{\infty} n(1-p)^{n-1} = \frac{1}{p^2} \quad (10)$$

となる。これを (4) 式に代入しなおすことで

$$E[Z] = \frac{\delta}{p} \quad (11)$$

が得られる。これにより題意は示された。

レポート課題 1.3

10 面サイコロの 0-9 の目をすべて出すために必要なサイコロを振る回数の平均を求める。

サイコロの目をコンプリートする回数を計算することは上記課題 1.2 で行ったタイムスロットの到着間隔の期待値を考えると同じで、 k 個目のサイコロの目が m 回目揃う確率は $(\frac{k}{10})(1 - \frac{k}{10})^{m-1}$ となる。これを m が無限大になるまで和を求めればよい。サイコロを振る回数の平均 E は

$$E = \sum_{k=1}^{10} \frac{1}{1 - \frac{k}{10}} \quad (12)$$

$$= 10(\frac{1}{10} + \frac{2}{10} + \frac{3}{10} + \dots + 1) \quad (13)$$

となりしたがっておおよそ $E = 29.28$ という値が得られる。

2.1 レポート課題 2.1

指数分布の密度関数が $f(x) = \lambda e^{-\lambda x}$ であることを用いて分布関数が $F(x) = 1 - e^{-\lambda x}$ となることを示す。

密度関数を定義されるすべての x で積分することで分布関数が求まるので計算をすると

$$F(x) = \int_0^x \lambda e^{-\lambda t} dt \quad (14)$$

$$= [-e^{-\lambda t}]_0^x \quad (15)$$

$$= 1 - e^{-\lambda x} \quad (16)$$

となり、よって題意は求まった。

2.2 レポート課題 2.2

期待値は確率とそのときの値を掛け合わせたものの総和で求まる。指数分布は連続な確率関数なので

$$E[x] = \int_0^\infty x(\lambda e^{-\lambda x}) dx \quad (17)$$

$$= [xe^{-\lambda x}]_0^\infty - \lambda \int_0^\infty -\frac{1}{\lambda} e^{-\lambda x} dx \quad (18)$$

$$= -\frac{1}{\lambda} [e^{-\lambda x}]_0^\infty \quad (19)$$

$$= \frac{1}{\lambda} \quad (20)$$

と部分積分などを用いることで求まった。

2.3 レポート課題 2.3

ここでは分布関数 $F(x) = 1 - e^{-\lambda x}$ の逆関数が $F^{-1}(x) = -\frac{1}{\lambda} \log(1 - x)$ となることを示す。

$$y = F(x) \quad (21)$$

$$= 1 - e^{-\lambda x} \quad (22)$$

とする。 x と y を入れ替えることで逆関数となる。入れ替えた後の関数を y について解いていく。

$$x = 1 - e^{-\lambda y} \quad (23)$$

$$e^{-\lambda y} = 1 - x \quad (24)$$

$$-\lambda y = \log(1 - x) \quad (25)$$

$$y = -\frac{1}{\lambda} \log(1 - x) \quad (26)$$

このように式を変形させることで示される。

2.4 レポート課題 3

ランダムな到着とみなせる現象はレジの行列や自動販売機に支払われる硬貨などが挙げられる。

レジの行列はとてもポピュラーな例である。客がその任意のタイミングでレジ行列に並び、レジでの会計が終わることでその処理が終了するため、ランダムな到着とみなせる。

自動販売機に支払われる硬貨も、その利用者が支払う際に自身が持つ任意の硬貨の組み合わせで支払いを行うため、硬貨によるある程度の重み (例えば 100 円硬貨での支払いは多く、50 円硬貨での支払いは少ない) がつくと予想されるが、ランダムな到着とみなすことができる。

3 実験

以下に実験とその結果について示していく。

3.1 実験 1

実験 1 では、間隔 δ により短く区切られた一つのタイムスロットでそれぞれ確率 p の確率的施行を行い、成功か失敗かを定める、ベルヌーイ過程を実験的に行う。

今回の実験では $\delta = 1, p = 0.1$ として計算をおこなう。この施行ののち、到着までいくつのタイムスロットで失敗をしたのかの到着間隔をヒストグラムにまとめる。

実験には Python を用いて発生させた乱数が 0.1 より小さいときに成功、大きいと失敗とし失敗した場合はまた同じ施行を繰り返させ、施行が失敗した回数を数えることでそれを到着間隔とした。

用いたコードは以下ソースコード 1 に示す。

ソースコード 1 Beenoulli.py

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import random
4
5 LOOP = 10 ** 6
6 P = 0.1
7 counter = []
8
9 for i in range(LOOP):
10     count = 1
11     while random.random() > P:
12         count += 1
13     counter.append(count)
14
```

```

15 hlist = np.histogram(counter, range=(1,50), bins=49, density=False)
16 # print(hlist[0]) #各到着間隔の個数リスト
17 # plt.figure()
18 # plt.hist(counter, range=(0, 50), bins=50) #到着間隔のヒストグラム
19 # print(hlist[0]/LOOP) #hlist を相対度数(確率)に計算したやつ
20 plt.figure()
21 plt.xlabel('ArrivalInterval')
22 plt.bar(np.arange(1,50), hlist[0]/LOOP, width=1.0) # 相対度数のヒストグラム
23 plt.show()

```

このコードによって得られたヒストグラムを図 1 に示す。

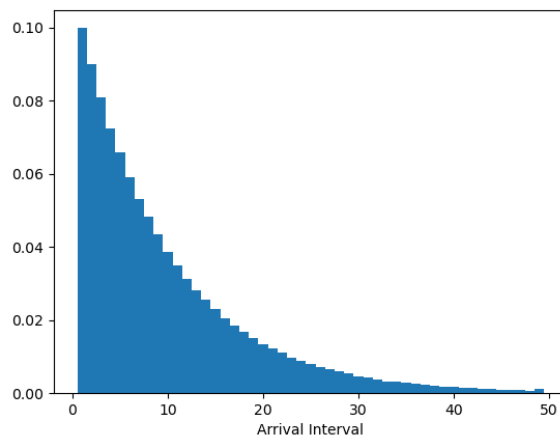


図 1 到着間隔の相対度数分布

この図 1 が到着間隔の実験的な分布となる。

3.2 実験 2

実験 2 では指数分布に従う乱数を実験的に発生させ、その累積相対度数のヒストグラムと平均値を求め、指数分布の分布関数や期待値と比較していく。

一様乱数から任意の分布に従う確率変数を生成するには、今必要としている確率変数の分布関数を $y = F(x), [0, 1)$ の一様乱数を U としたとき

$$X = F^{-1}(U) \quad (27)$$

を計算すればよい。

レポート課題 2.3 で示した内容を用いれば、指数分布を一様乱数から生成するための式は

$$X = -\frac{1}{\lambda} \log(1 - U) \quad (28)$$

となる。この式から得られた乱数を統計的に処理して累積相対度数のグラフと平均値を取得する。
用いたコードは以下ソースコード 2 に示す。

ソースコード 2 Exporand.py

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import random as rand
4 import itertools
5
6 LOOP = 10 ** 6
7 lamb = 1 # λの値,変更してもよい
8 maxnum = 15 # グラフの最大出力,λを変えたときに変えるかも
9
10 randnum = [(-np.log(1-rand.random()))/lamb for _ in range(LOOP)]
11 # 指数分布に従う乱数のリスト
12
13 hlist = np.histogram(randnum, range=(0,maxnum), bins=maxnum * 10, density=
    False)
14 # 乱数を 0.1ずつの区分に分けて個数を計算
15
16 # print(hlist[0]) #試しに出力
17
18 plt.figure() # 相対頻度のグラフ
19 plt.hist(randnum, range=(0, maxnum), bins=maxnum * 10)
20
21 # print(np.array(list(itertools.accumulate(hlist[0])))/LOOP) #試しに出力
22 print('平均:' + str(np.average(randnum))) # 平均値出力
23 print('期待値:' + str(1/lamb)) # 理論値平均値出力
24
25 x = np.arange(0,maxnum,0.1)
26 plt.figure() # 累積相対度数のグラフ
27 plt.xlabel('Generated_Random_Numbers')
28 plt.ylabel('Cumulative_Frequency')
29 plt.legend()
30 plt.plot(x,1-np.exp(-lamb*x),color='r', label='Theoretical_function')
31 plt.bar(np.arange(0,maxnum,0.1), np.array(list(itertools.accumulate(hlist
    [0])))/LOOP, width=0.1, label='Experimental_value')
32 # hlist の累積和を計算して numpy の配列に変更
33 plt.show()
```

このコードで得られたヒストグラムを図 2 に示す。

図 2 では青色のヒストグラムが実験で発生させた乱数の累積グラフ、赤色のグラフが指数分布の

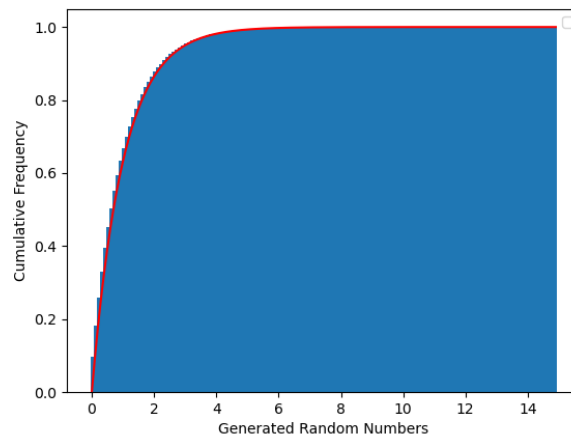


図2 指数分布に従う乱数の累積相対度数と指数分布の分布関数

分布関数になっている。どちらもパラメータ λ は 1 で計算している。ここからわかるように、ほとんど一致したグラフを生成できていることがわかる。

またこの施行で得られた平均値は 0.999425708871489 となった。期待値は $\frac{1}{\lambda}$ であることから $\lambda = 1$ であるとき期待値も 1 となるため、そこそこの精度で一致していることが読み取れる。

3.3 実験3 - 結果

実験3ではパラメータ λ_1 のポアソン過程に従って到着するバスに、パラメータ λ_2 のポアソン過程に従って到着する乗客が乗るようなバスの運行をシミュレートしていく。条件として到着し、バスが来るまで待っている乗客はすべて到着したバスに乗るものとしてシミュレートする。

またパラメータ λ_1, λ_2 はその都度適宜変更していく。

実験ではソースコード3を用いた。

ソースコード3 Buses.py

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import random as rand
4
5 LOOP = 10 ** 6
6 lam_1 = 2 # バスの到着間隔 単位時間にlam_1 台
7 lam_2 = 5 # 乗客の到着間隔 単位時間にlam_2 人
8
9 cus_list = [] # 各バスに乗る乗客の人数
10 time_list = [] # すべての乗客がバスを待つ時間
11
12 bus_delta = (-np.log(1-rand.random()))/lam_1
```



```

13 # バスの到着間隔のリスト
14 cus_del_sum = (-np.log(1-rand.random()))/lam_2 # 乗客の到着間隔
15
16 for i in range(LOOP):
17     cus_sum = 0 # 乗客の合計
18     while(cus_del_sum < bus_delta):
19         # 乗客の到着間隔の合計がバスの到着間隔を超えるまで
20         cus_sum += 1 # バスの乗客を 1人追加
21         time_list.append(bus_delta - cus_del_sum) # 待ち時間をリストに追加
22         cus_del_sum += (-np.log(1-rand.random()))/lam_2 # 乗客をもう一人並ばせる
23     cus_list.append(cus_sum)
24     bus_delta += (-np.log(1-rand.random()))/lam_1
25
26 pass_list = np.array([item for item in cus_list for _ in range(item)])
27
28 hlist = np.histogram(cus_list, range=(0,35), bins=35, density=False)
29 phlist = np.histogram(pass_list, range=(0,45), bins=45, density=False)
30 thlist = np.histogram(time_list, range=(0,4), bins=400, density=False)
31
32 print('単位時間に'+str(lam_1)+'台 ( $\lambda \sim e^{2^{82^{81}}}$ )')
33 print('単位時間に'+str(lam_2)+'人 ( $\lambda \sim e^{2^{82^{82}}}$ )')
34 print('乗客の平均:'+str(np.average(cus_list)))
35 print('同乗者の平均:'+str((sum(pass_list))/sum(cus_list)))
36 print('待ち時間の平均:'+str(np.average(time_list)))
37
38 plt.figure()
39 plt.xlabel('バスに乗る乗客 (人)', fontname="MS_Gothic")
40 plt.ylabel('相対度数', fontname="MS_Gothic")
41 plt.bar(np.arange(0,35), hlist[0]/LOOP, width=1.0)
42 x = np.arange(0,35)
43 plt.plot(x, (lam_1/(lam_2))*np.exp(-(lam_1/(lam_2))*x), color='r')
44
45 # plt.figure()
46 # plt.xlabel('乗客から見た同乗者 (人)', fontname="MS Gothic")
47 # plt.ylabel('相対度数', fontname="MS Gothic")
48 # plt.bar(np.arange(0,45), phlist[0]/LOOP, width=1.0)
49
50 # plt.figure()
51 # plt.xlabel('乗客の待ち時間 (時間)', fontname="MS Gothic")
52 # plt.ylabel('相対度数', fontname="MS Gothic")
53 # plt.bar(np.arange(0,4,0.01), thlist[0]/LOOP, width=0.01)
54 plt.show()

```

先にソースコード 3 を実行して得られた乗客者数、同乗者数、乗客の待ち時間の平均をいくつかの λ で求めた結果を表 1 に示しておく。

表 1 実験 3 で得られた平均値

λ_1	λ_2	乗客者数平均	同乗者数平均	待ち時間平均
1	5	5.00864	10.97850	1.00119
2	5	2.49612	5.99988	0.49941
1	10	9.99268	20.96280	0.99802
2	10	5.01614	11.05486	0.50255

3.3.1 実験 3.1

実験 3.1 ではバスの平均乗客者数とその乗客者数の分布について調べていく。

ソースコード 3 上では、平均乗客者数と乗客者数の分布を実験的に調べるために while 文を用いた。while 文では脱出条件として、次に来るバスについて、それまでのバスの到着間隔の累積和よりもこれまで来た乗客の到着間隔の累積和が大きくなる時に脱出する、すなわちバスが次に来た段階で乗客のカウントをリセットさせている。逆に脱出条件に達していないときはバスの乗客のカウントを一つ増やし、次に来る乗客の到着間隔を新たに乱数で生成している。

この乗客者数のカウントを配列にそれぞれ保存していき、その配列の頻度分析をすることで乗客者数の分布を、平均を求めることで乗客者数の平均をそれぞれ求めていく。

ソースコード 3 で得られた乗客者数の相対度数分布のグラフを以下図 3,4 に示す。

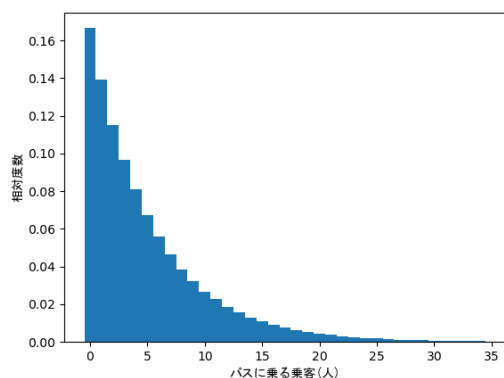


図 3 $\lambda_1 = 1, \lambda_2 = 5$ のときの乗客者数ヒストグラム

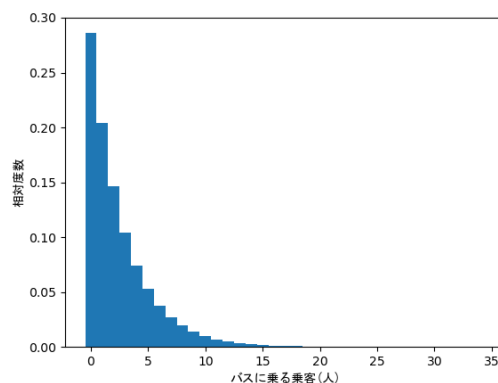


図 4 $\lambda_1 = 2, \lambda_2 = 5$ のときの乗客者数ヒストグラム

乗客者数の平均は表 1 に示してあるが、再度乗客者数の平均だけ表 2 に示す。

図 3,4 を見ると指数分布のような度数分布として出力されている。表 2 を見ると乗客者数平均と λ にある関係が見える。

表2 実験3で得られた乗客者数平均値と $\frac{\lambda_2}{\lambda_1}$ の値

λ_1	λ_2	乗客者数平均	$\frac{\lambda_2}{\lambda_1}$
1	5	5.00864	5
2	5	2.49612	2.5
1	10	9.99268	10
2	10	5.01614	5

小数誤差を考慮し整理すると、乗客者数平均を C_{ave} としたとき

$$C_{ave} = \frac{\lambda_2}{\lambda_1} \quad (29)$$

となることが考えられる。

また前述したように図3,4では指数分布のような度数分布になったが実際に期待値が C_{ave} となるような指数分布を図3に重ねると図5のようになった。

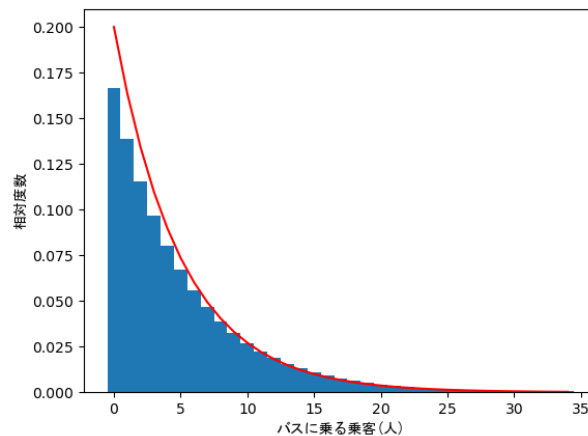


図5 図3に $\frac{1}{C_{ave}}$ がパラメータとなる指数分布を重ねたヒストグラム

この図5を見ると赤線で示される期待値が C_{ave} となる密度関数と、実験で得た度数分布がかなり類似した分布になっていることが分かる。この分布の類似は λ を別の値に変えても確認された。

3.3.2 実験3.2

実験3.2ではバスの中の乗客から見た混み具合、すなわち乗客から見たバスの同乗者数の分布とその平均値を求めていく。

ソースコード3内では、同乗者数というものを、バスの乗客全員から見た、そのバスに乗っている、自分を含めた同乗者をすべてカウントすると解釈して実装した。すなわち同乗者数は乗客者数の2乗でカウントされる。前実験で用いた乗客者数のリストをPythonのリスト内包表記を二重で

用いることで配列のすべての要素一つ一つの要素数をその要素の 2 乗に増やしている。この配列を numpy に付属する関数で解析して度数分布を、乗客者数で割ることで乗客から見た同乗者数の平均を求めている。

ソースコード 3 を実行することで得られた同乗者数の相対度数分布のグラフを以下図 6,7 に示す。

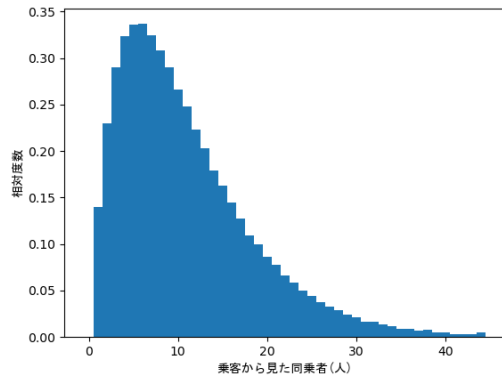


図 6 $\lambda_1 = 1, \lambda_2 = 5$ のときの同乗者数ヒストグラム

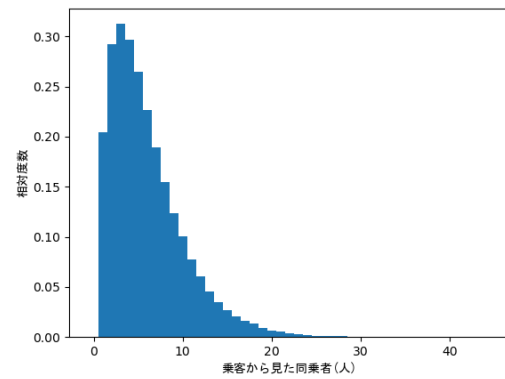


図 7 $\lambda_1 = 2, \lambda_2 = 5$ のときの同乗者数ヒストグラム

同乗者数の平均も表 1 に示してあるが、再度同乗者数の平均を表 3 に示す。

表 3 実験 3 で得られた同乗者数平均値と $2\frac{\lambda_2}{\lambda_1} + 1$ の値

λ_1	λ_2	同乗者数平均	$2\frac{\lambda_2}{\lambda_1} + 1$
1	5	10.97850	11
2	5	5.99988	6
1	10	20.96280	21
2	10	11.05486	11

図 6,7 を見ると χ 二乗分布のようなヒストグラムに見えるが、実際、一様分布ではない乱数の二乗和の分布となるので χ 二乗分布の密度関数に近づくことも考えられる。ただ χ 二乗分布ではない。

表 3 から同乗者数の平均と λ の値の関係を考えると同乗者数の平均値を P_{ave} とすると

$$P_{ave} = 2\frac{\lambda_2}{\lambda_1} + 1 \quad (30)$$

となることが考えられる。実際にこの式に λ を代入して計算した値を表 3 に示している。

3.3.3 実験 3.3

実験 3.3 ではバスの乗客がバス停に到着してからバスが着くまでに待つ時間の分布とその平均値を求めていく。

ソースコード 3 内では、乗客が乗るバスが到着した時間からその乗客が到着した時間を引くことで待ち時間を求めている。この待ち時間を配列に追加していくことで、その配列を解析して度数分布を、その配列の平均値をとることで待ち時間の平均を求めている。

ソースコード 3 を実行することで得られた同乗者数の相対度数分布のグラフを以下図 8,9 に示す。

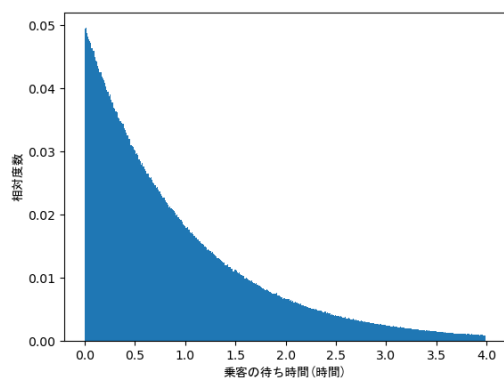


図 8 $\lambda_1 = 1, \lambda_2 = 5$ のときの待ち時間ヒストグラム

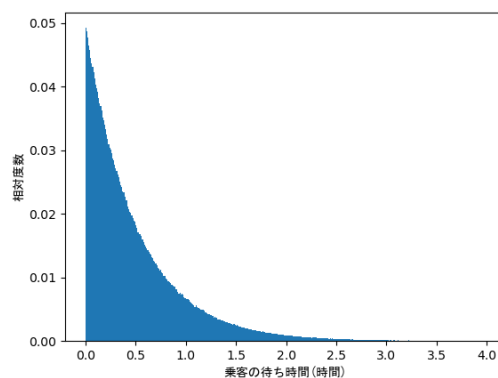


図 9 $\lambda_1 = 2, \lambda_2 = 5$ のときの待ち時間ヒストグラム

同乗者数の平均も表 1 に示してあるが、再度乗客の待ち時間の平均を表 4 に示す。

表 4 実験 3 で得られた乗客の待ち時間の平均値と $\frac{1}{\lambda_1}$ の値

λ_1	λ_2	乗客の待ち時間の平均	$\frac{\lambda_2}{\lambda_1}$
1	5	1.00119	1
2	5	0.49941	0.5
1	10	0.99802	1
2	10	0.50255	0.5

図 8,9 を見ると指数分布のようなヒストグラムに見える。

表 4 から乗客の待ち時間の平均と λ の値の関係を考えると乗客の待ち時間の平均値を T_{ave} とすると

$$T_{ave} = \frac{1}{\lambda_1} \quad (31)$$

となることが考えられる。実際にこの式に λ を代入して計算した値を表 3 に示している。

3.4 実験 3 - 考察

以下で実験 3 の結果について考察していく。

3.4.1 考察 1

考察 1 ではバスの平均乗客数と乗客の平均同乗者数を比較し、関係を考える。先の実験 3.1, 3.2 においてそれぞれの平均値の理論式を

$$C_{ave} = \frac{\lambda_2}{\lambda_1} \quad (32)$$

$$P_{ave} = 2\frac{\lambda_2}{\lambda_1} + 1 \quad (33)$$

と推定した。

この二式より関係は

$$P_{ave} = 2C_{ave} + 1 \quad (34)$$

とすることができる。この (34) 式がどのように導出されているのかについて考察していく。ただここでは C_{ave} の理論的な導出は易しいものではないことが予想されるので経験則に基づく式にはなるがこのまま用いる。

この導出を考える前に P_{ave} をどのように求めたかに立ち返って考える。実験では乗客数数の二乗和を乗客数数で割った値として計算した。これはすなわち乗客数数のある密度関数 $f(x)$ に従う確率変数 X としたとき、+1 の項を無視すれば

$$P_{ave} = \frac{\sum X^2}{\sum X} \quad (35)$$

を計算していることになる。

ここでこの分数をバスの台数となる n で分母分子割ると

$$P_{ave} = \frac{\frac{\sum X^2}{n}}{\frac{\sum X}{n}} \quad (36)$$

となる。ここでこの分母 $\frac{\sum X}{n}$ は乗客数数の総和をバスの台数で割ったもの、すなわちバス一台あたりの乗客数数 C_{ave} と同じになる。ということで式は

$$P_{ave} = \frac{\frac{\sum X^2}{n}}{C_{ave}} \quad (37)$$

と変形できる。

ここで分子に注目していく。先の実験 3.1 で図 5 のように、乗客数数の分布は平均値が C_{ave} となる指数分布によく似ているという指摘をした。これを参考に X が実際に平均値が C_{ave} となる指数分布に従う確率変数だと仮定すると式 (37) の分子は指数分布の確率変数の二乗平均となり、

これは実際に計算が可能である。詳細な計算方法は省く [1] が指数分布における二乗平均は平均値が $E[x]$ のとき $2E[x]^2$ となることが知られている。これをそのまま式 (37) に代入することで

$$P_{ave} = \frac{2C_{ave}^2}{C_{ave}} \quad (38)$$

$$= 2C_{ave} \quad (39)$$

と計算でき、+1 の項を無視してしまっているがおおよそ関係が導出できたのではないかとおもう。

3.4.2 考察 2

考察 2 では乗客の平均待ち時間とバスの平均間隔の関係について考察していく。

バスの平均間隔はパラメータ λ_1 の指数分布を用いてバスの到着間隔を求めているため、理論値として $\frac{1}{\lambda_1}$ となる。乗客の平均待ち時間は実験 3.3 の結果より類推して経験則的に $\frac{1}{\lambda_1}$ であると予測した。

この二つの式は同じ式となっていることが分かる。この二式がどのように同じになるのかは単純で、 λ_2 が関係してこない事象のため乗客の到着間隔を単位時間に固定して考えると、レート λ_1 のポアソン過程に従うバスの到着から待ち時間を考えることはバスの到着間隔を求めていることと変わらないことが分かる。

したがってこの二式が同じものとなることが分かる。

参考文献

- [1] 指数分布の期待値・分散の導出（証明） — AVILEN AI Trend <https://ai-trend.jp/basic-study/exponential-distribution/e-parameter-derivation/> 2023/11/23 閲覧