

ATM と待ち行列

M.Shimura

JCD02773@nifty.ne.jp

2003 12 13

JAPLA2003

1 待ち行列 queue

ATM と待ち行列のシミュレーション。ATM の稼働率、顧客の待ち時間（平均、最大）を求める。

郵便局や銀行の窓口・ATM、高速道路の料金所、スーパーやコンビニのレジ、チケット売り場、公衆トイレなどであるサービスを受けるために、客が作る行列を待ち行列 (queue) という。この待ち行列を分析し、シミュレーションを行う。

1.1 ATM と待ち行列の例

No 到着時間 (Sec) サービス時間 (Sec)

1	25	40	13	37	60	26	28	99	39	143	202
2	170	37	14	88	190	27	1	184	40	130	317
3	101	168	15	121	72	28	48	115	41	11	232
4	308	58	16	76	58	29	37	81	42	26	24
5	27	89	17	201	48	30	232	70	43	39	92
6	2	52	18	74	64	31	1	372	44	22	318
7	31	60	19	91	46	32	6	103	45	45	33
8	80	59	20	14	73	33	7	48	46	62	17
9	29	133	21	59	124	34	54	65	47	77	36
10	16	42	22	28	143	35	1	79	48	104	86
11	21	113	23	1	81	36	64	45	49	69	131
12	16	27	24	4	65	37	1	55	50	104	54
			25	43	64	38	2	65			

出典 東北大学統計グループ「これだけは知っておこう 統計学」 P 2 1 4

1.2 Script list

提示された到着時間とサービス時間のデータをもとに、対数変換して、指数一様分布と対数分布を求め、シュミレーションを行う場合に加え、到着時間とサービス時間の平均と分散をもとに、シュミレーションを行う。

シュミレーションは、一回ごとに結果を出すようにしてある。ループを加えれば、反復は可能である。

machi_0	提示されたデータの平均、分散を計算し、サービス時間を対数変換した後、サービス時間の対数平均、対数分散をクラメール法により求める	machi_0 y. (data)
machi_00	提示されたデータの平均、分散（実データでも仮想データでも可）から、サービス時間の対数平均、対数分散をクラメール法により求める	machi_00 y. (data)

machi_sim0	乱数によりメインの待ち行列を作成する。乱数により指数一様分布と対数分布を求め、到着時間とサービスタイムをシュミレートする。	x. mach_sim0 y. x. 乱数の数 (客数) (not add Machine nr.) y. data machi_00 を使うときは、平均の分散のマトリクス
machi_sim1	指定台数の仮想マシンを設定し、到着時間とマシンの空きを比較しながら客ごとのサービスの開始と終了時間を計算するメインルーチン。	x. mach_sim1 y. x. 乱数の数 (客数) マシン台数 List e.g. 20 2 y. data e.g. 20 3 machi_sim1 tdata
machi_anal	到着からサービス開始までに待たされた客のリストと各待ち時間を計算する。	x. machi_anal y. x. y. is same as machi_sim1
wait_number	待っている人の構成を調べる。 99999 を挟まない数字が連続すれば、待ち人数 e.g. 5 6 7 is max 3	wait_number '' '' is null
count_number	待ち行列の機会毎の待ち人数を数える。	count_number ''
machi_main	メインプログラム	x.(a b) machi_main y. a b is list a 乱数発生個数 b 装置数 y. データ または平均、分散の行列 (machi_00) に書き換え

plot_random	乱数を昇順ソートしてXの値ごとの個数をカウントしグラフを書く。(Xは10倍してある)	plot_random y.
csort	補助関数 sort	
neben_tindex	補助関数 TINDEX の一段上との差分を取る	
rnd run rn rno	N.Thomson の乱数スクリプト	
exp_h	補助関数 expand like APL Matrix を指標により拡張する	

1.3 指数分布と対数分布

到着時間 サービス時間 対数サービス時間の計算を行う

	到着時間	サービス時間	対数サービス時間
合計	2952	4889	217.448
平均	59.54	97.78	4.34897
分散	4019.41	5895.53	0.43661

到着時間

$$\frac{F(t+dt) - F(t)}{1 - F(t)} = \frac{e^{-\lambda t} - e^{-\lambda(t+dt)}}{e^{-\lambda t}} = 1 - e^{-\lambda dt}$$

現時点から dt 時間の中に次の客が到着する確率は、前の客の到着とは関係なく、常に $1 - e^{-\lambda dt}$ になる。

$$\frac{1}{\lambda} = 59.5$$

$$1\%59.5$$

$$0.016806$$

客の到着時間は、 $\lambda = 0.0168$ の指数分布に従う。

サービス時間

対数分布で、確率変数 $Y = \ln X$ 、平均 μ 、分散 σ^2 の正規分布に従うとき、サービス待

ち時間の平均と分散は次のようになる。

対数分	$f(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\ln X - \mu)^2}{2\sigma^2}}$	
平均	$e^{\mu + \frac{\sigma^2}{2}} (= e^{2\mu + \sigma^2})$	97.78
分散	$e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2}$	5895.53

$$\begin{cases} e^{\mu + \frac{\sigma^2}{2}} = 97.78 \\ e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2} = 5895.53 \end{cases}$$

$e^{2\mu + 2\sigma^2} = 58.95 + e^{2\mu + \sigma^2}$ 平均 両辺に対数をとると

$$\begin{cases} 2\mu + \sigma^2 = 9.16544 \\ 2\mu + 2\sigma^2 = 9.64578 \end{cases}$$

この連立方程式を解くと、対数分布の平均と分散が求まる。生データがない場合は、この式により求める。

$$a = 2 * \ln 97.78 = 2\mu + \sigma^2$$

$$2 * \ln 97.78$$

$$9.16544$$

$$e^{2\mu + 2\sigma^2} = e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2} + e^{2\mu + \sigma^2} = 5895.78 + e^a$$

$$5895.53 + 1 * e^{2 * \ln 97.78}$$

$$15456.5$$

$$e^{2 * \ln 97.78} = 1 * e^{2 * \ln 97.78}$$

$$9.64578$$

連立方程式をクラメル法で解く

$$\text{dat} = \begin{bmatrix} 2 & 1 & 9.16544 & 2 & 2 & 9.64578 \\ 2 & 2 & 9.64578 & 1 & 1 & 9.16544 \end{bmatrix}$$

dat

$$2 \ 1 \ 9.16544$$

$$2 \ 2 \ 9.64578$$

cr = %}:"1

cr dat

1 _1.11022e_15 4.34255
 1.77636e_15 1 0.48034

客の到着時間は、 $\lambda = \frac{1}{59.5} = 0.0168$ の指数分布に従う。

客のサービス時間は平均 4.34255 分散 0.48034 の対数分布に従う。

到着時間	指数分布の一様乱数	$\lambda = \frac{1}{59.5} = 0.0168$	u は一様乱数 $F(T) = 1 - e^{-\lambda T}$ $t = F^{-1}u = -\frac{1}{\lambda} \ln(1 - u)$ $u) = -\frac{1}{\lambda} \ln(u)$
サービス時間	対数分布の正規乱数	平均 4.34255 分散 0.48034	$s = \sigma t + m$

1.3.1 一様乱数と正規乱数

一様乱数を打ち出す関数は、多くの言語が、内部関数やパッケージに持っている。

$N(0, 1)$ の正規分布を求める関数は、パーセント点での積分値をもとめるものが多いが、乱数は意外と見受けられない。

1.3.2 指数正規分布

指数正規分布は次により求められる。 $t = -\frac{1}{\lambda} \ln(u)$

```
plot_random -(% LMDA)* ^ . rnd 10000
```

1.3.3 対数分布

対数分布は、標準分布 $N(0, 1)$ に従う乱数を発生し、

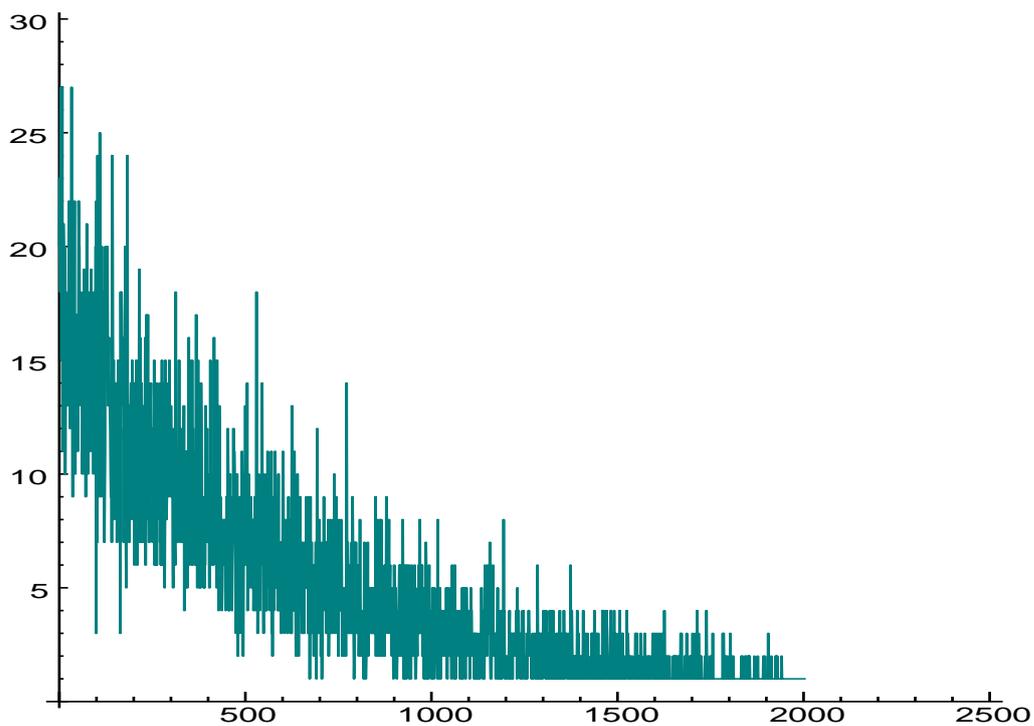


図 1: exponential distribution

$$s = \sigma t + m$$

$$e^s$$

を求める。

N.Thomson の著書に、任意の平均と標準偏差を与えて、その構成での正規分布を求める関数が紹介されている。これを用いて、平均0 標準偏差1の正規分布を求め、上の式により対数分布に変換する。^{Λ1}

```
plot_random ^ 4.3490 + (%: 0.4366)* rno 0 1 10000
```

^{Λ1} 正規分布ではマイナスが現れ複素数がでる。

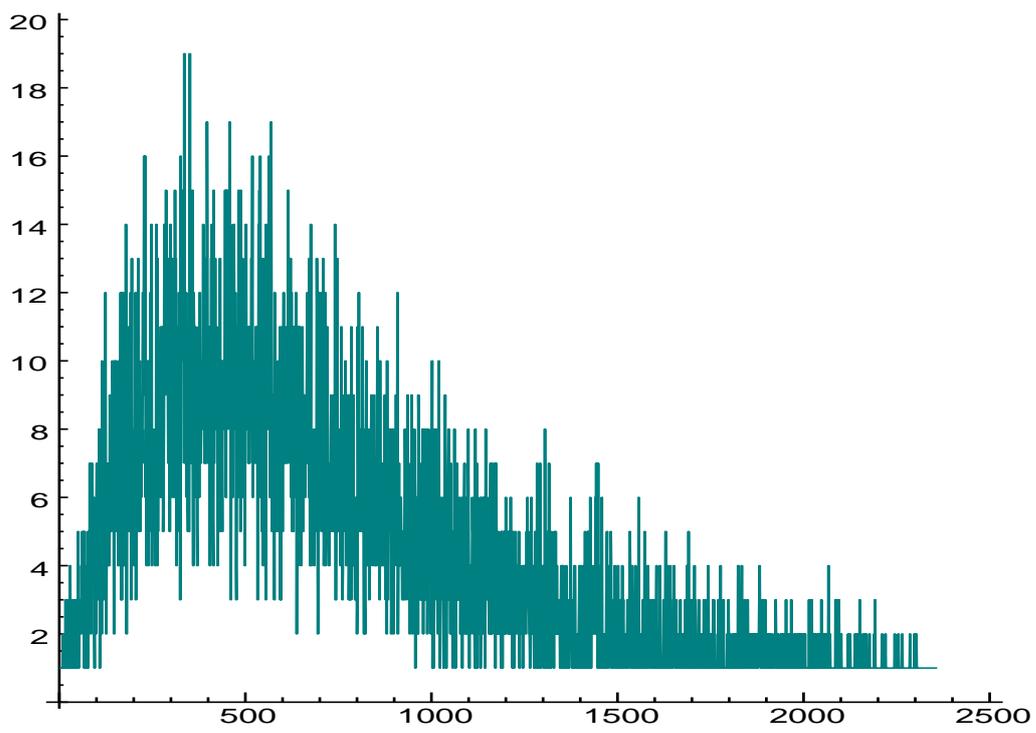


图 2: In normal distribution

1.4 経過と解説

1.4.1 machi_0 machi_00

サービス時間の平均の分散が対数分布の平均と分散になるように、パラメータを定める。machi_0 はデータから算定する方法、machi_00 は到着時間とサービス時間の平均と分散を与え、ケース毎のシミュレーションができるようにした。

```

M01=: y.,.^ . 1 sel y.      NB. add ln arrive time
LMDA=: %(0 { M02=: mean M01) NB. lambda
M04=: mean M01              NB. mean with ln
M05=: var M01               NB. var with ln
A=: +: ^ . 1{M04            NB. e^(m+(s^2)/2)
B=: ^ . (1{M05) + 1x1 ^ A   NB. e^(2m+2s^2)-e^(2m+s^2)
MAT=: 2 3 $ 2 ,1,A,2, 2 ,B NB. mat for cramer
M06=: cr MAT                NB. Theorem value
)

```

平均	$e^{\mu + \frac{\sigma^2}{2}}$	97.78
標準偏差	$e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2}$	5895.53

$$\begin{cases} 2\mu + \sigma^2 = 9.16544 \\ 2\mu + 2\sigma^2 = 9.64578 \end{cases}$$

$$A = 2\mu + \sigma^2 = 2 \times \ln 97.18$$

A サービス時間の平均を対数変換して2倍する。

```

+: ^ . 1{M04
9.16544

```

$$B = e^{2\mu + 2\sigma^2} - e^{2\mu + \sigma^2} = \sigma^2 + e^A$$

$$2\mu + 2\sigma^2 = \ln(\sigma^2 + e^A)$$

```

B=: ^ . (1{M05) + 1x1 ^ A

```

```

9.64578

```

$$\begin{cases} 2\mu + \sigma^2 = 9.16544 \\ 2\mu + 2\sigma^2 = 9.64578 \end{cases}$$

MAT

```
2 1 9.16544
2 2 9.64578
```

`cr=: %.`:"1 NB. Cramer method

連立方程式の解法には後に、特異な結果が出たときに、行列に問題がなかったか確認できるクラメル法を用いているが、連立方程式の解法は特にこだわる必要はない。

`machi_00` は

到着時間の平均 `a` サービス時間の平均 `b` のマトリクスで与える。

到着時間の分散 `c` サービス時間の分散 `d`

`foo=: 2 2 $ a b c d` で入力する。

ここで、分散は `n` で割っており、`n-1` では行っていない。東北大学の例題も `n` 型である。最近は `n` 型も多い。

1.4.2 machi_sim0

```
machi_sim0=: 4 : 0
```

NB. `x.` nr. of customers (not need nr. of mashine)

NB. `y.` data (or `machi_00` mean and var matrix)

NB. Usage: `x. machi_sim0 y.`

NB. calc arrival time and service time

NB. random with mean and sd

NB. `t = rno (mean 0 Var 1 nr. x.)// and e ^ m + sigma t`

```
M10=: {: "1 M06
```

```
M11=: 1x1 ^S=: ({. M10) + (%: {: M10)* rno 0,1,x.
```

```
M13=: -(% LMDA)* ^ . M12=: rnd x. NB. -1/lambda ln u
```

```
M12, .M13, .(>+/ L:0 <\ M13), .S, .M11
```

```
)
```

M10 対数サービス時間の平均と分散を受ける。machi_0, machi_00 のどちらでもよい。最終的には machi_main で指定する。

NB. M11=: 1x1 ^S=: ({: M04) + (%: {: M05)* rno 0,1,x.

M11 rno 0,1,x. で平均0、分散1の正規分布をx.個打ち出し,s = σt + m, e^Sを求める。

:%: {:M05 σ is $\sqrt{\text{var}}$

{:M04 m mean

M13=: -(% LMDA)* ^ . M12=: rnd x.

M13 $t = -\frac{1}{\lambda} \ln(u)$, u は [0, 1] 上の一様乱数

20 machi_sim0 tdat

一様乱数 到着時間 累計 N(4.345,0.480) サービス時間

```
-----
M12,.M13,.(>+/ L:0 <\ M13),.S,.M11
0.907386 5.78654 5.78654 4.1939 66.2805
0.434714 49.6009 55.3874 5.5135 248.018
0.230527 87.3682 142.756 4.8794 131.552
0.470712 44.864 187.62 4.4314 84.049
0.248754 82.8374 270.457 4.62937 102.45
0.810747 12.4915 282.948 4.82388 124.447
0.219152 90.3811 373.33 4.63104 102.62
0.289502 73.8055 447.135 4.83327 125.621
0.652955 25.3788 472.514 3.33707 28.1366
0.208676 93.2974 565.811 3.72073 41.2944
0.225699 88.6285 654.44 4.2445 69.7211
0.317195 68.3662 722.806 4.27584 71.9403
0.0913529 142.481 865.287 4.81587 123.454
0.368418 59.4528 924.739 5.20321 181.855
0.00790594 288.182 1212.92 3.70372 40.5979
0.875103 7.94343 1220.86 3.60118 36.6416
```

```

0.861796 8.85581 1229.72 6.59391 730.633
0.198351 96.3188 1326.04 4.27725 72.0422
  0.69123 21.9871 1348.03 4.17725 65.1864
0.495345 41.8269 1389.85 5.11963 167.274

```

乱数による到着時間とサービス時間が求めれば、実際に、ATMなどの装置をセットして、カウントすればよい。

到着時間とサービス時間は、個別の乱数を用いている。

1.4.3 machi_sim1

NB. Usage: x. (a b) machi_sim1 y.

左引数の最初は打ち出す乱数の数（客数）次の引数は装置の数である。リスト（ベクトル）で与える。

e.g. 20 2 machi_sim1 tdata

仮想装置 x. で与えた台数の仮想機械をセットし、装置数の最初の x. 人のデータを入れて、ソートして最初に一番早く終わる客（装置）を置く。到着は一人ずつの逐次処理なので、この最初の装置の稼働、空きをカウントして、空いていればサービスを開始し、稼働していれば、そのサービスの終了時間をサービス開始時間とする。最初から最終顧客の到着まで逐次処理し、その顧客の装置の終了まで最終処理を行う。

```

MACHINE=:/:~ ,{:("1) X2 { . M15 NB. input data to Mashine
MACHINE=: /:~( END_TIME (0) } MACHINE) NB. Machine newer

```

以下がシュミレーションの基本テーブルである。

```

20 2 machi_sim1 tdat
No 到着 サービス所要時間累計 開始 終了
-----
T_MAT
0 5.78654 66.2805 72.0671 5.78654 72.0671
1 55.3874 248.018 303.405 55.3874 303.405
2 142.756 131.552 274.307 142.756 274.307
3 187.62 84.049 271.669 274.307 358.356

```

4	270.457	102.45	372.907	303.405	405.855
5	282.948	124.447	407.395	358.356	482.803
6	373.33	102.62	475.95	405.855	508.476
7	447.135	125.621	572.756	482.803	608.424
8	472.514	28.1366	500.65	508.476	536.612
9	565.811	41.2944	607.106	565.811	607.106
10	654.44	69.7211	724.161	654.44	724.161
11	722.806	71.9403	794.746	722.806	794.746
12	865.287	123.454	988.741	865.287	988.741
13	924.739	181.855	1106.59	924.739	1106.59
14	1212.92	40.5979	1253.52	1212.92	1253.52
15	1220.86	36.6416	1257.51	1220.86	1257.51
16	1229.72	730.633	1960.35	1253.52	1984.15
17	1326.04	72.0422	1398.08	1326.04	1398.08
18	1348.03	65.1864	1413.21	1398.08	1463.27
19	1389.85	167.274	1557.13	1463.27	1630.54

次の TINDEX も重要なファイルであり、各到着客のサービス開始時刻と、99999 で示した、サービス終了時刻を記憶しており、この2本のファイルを解析する事により、シュミレーションの解析を行う。

TINDEX	
0	5.78654
1	55.3874
99999	72.0671
2	142.756
3	187.62
4	270.457
99999	274.307
5	282.948
99999	303.405

1.4.4 machi_anal

各顧客の到着時刻とサービス開始時刻の差で、待ちの有無の判定を行う。

状態継続時間 TINDEX を neben_tindex で t_1 との差分を取り、状態継続時間とする。

```
M20
0 5.78654 49.6009
1 55.3874 16.6797
99999 72.0671 70.6885
2 142.756 44.864
3 187.62 82.8374
4 270.457 3.85028
99999 274.307 8.64119
```

待ち時間は、サービス開始時刻から到着時刻を引けばよい。M23 は待ちの出たケースのリストである。

```
M23
No Arrival Serv.time sum start finish wait time
-----
3 187.62 84.049 271.669 274.307 358.356 86.6877
4 270.457 102.45 372.907 303.405 405.855 32.9484
5 282.948 124.447 407.395 358.356 482.803 75.4078
6 373.33 102.62 475.95 405.855 508.476 32.5257
7 447.135 125.621 572.756 482.803 608.424 35.6681
8 472.514 28.1366 500.65 508.476 536.612 35.9618
16 1229.72 730.633 1960.35 1253.52 1984.15 23.7987
18 1348.03 65.1864 1413.21 1398.08 1463.27 50.0551
19 1389.85 167.274 1557.13 1463.27 1630.54 73.4146
```

1.4.5 wait_number

M25 待たなかったリスト

M30 M20 から待たなかったリストを抜く。この中から 99999 で区切られた区間で、顧客 NO が継続しているものが、待ち行列の長さ（客数）である。（3, 4 2人）これは、count_number でカウントする。

M30

```
99999 72.0671 70.6885
      3 187.62 82.8374
      4 270.457 3.85028
99999 274.307 8.64119
      5 282.948 20.4569
99999 303.405 54.9509
99999 358.356 14.9733
      6 373.33 32.5257
99999 405.855 41.2797
      7 447.135 25.3788
      8 472.514 10.2893
99999 482.803 25.6725
```

1.4.6 machi_main

入力の整理、関数の起動、出力の整理を行う。他の言語で、バッチに相当する。

20 2 machi_main tdat

No Arrive Serv.time sum start serv. end serv. wait time

```
+-----+
| 3 187.62 84.049 271.669 274.307 358.356 86.6877|
| 4 270.457 102.45 372.907 303.405 405.855 32.9484|
| 5 282.948 124.447 407.395 358.356 482.803 75.4078|
```

```

| 6 373.33 102.62 475.95 405.855 508.476 32.5257|
| 7 447.135 125.621 572.756 482.803 608.424 35.6681|
| 8 472.514 28.1366 500.65 508.476 536.612 35.9618|
|16 1229.72 730.633 1960.35 1253.52 1984.15 23.7987|
|18 1348.03 65.1864 1413.21 1398.08 1463.27 50.0551|
|19 1389.85 167.274 1557.13 1463.27 1630.54 73.4146|
+-----+
|9 49.6075 | NB. wait number of persons
+-----+ NB. and average time
|1.31735 | NB. rate of operating
+-----+
|2 1 1 2 1 2 | NB. waiting pattern
+-----+

```

待ち人数とパターン、平均待ち時間、装置稼働率を求めている。

装置稼働率は、0（シミュレーション開始）から最後の顧客に対するサービス終了じこくを取っているが、営業時間が決まっているなら、その時間を入れることになる。

1.5 Simulation

Usage: x. machi_main y.

x. a b (or a,b)

a number of random

b number of facilities

y. data matrix or mean var matrix

e.g. 20 2 machi_main tdata

一回ごとの結果を出力する。Box で表示しているので、このままではEXCELに入らない。Cut and Paste でワープロに張り付ける。

95%信頼区間は次式により求める。

$$P\left[\bar{x} - 1.96\frac{\sigma}{\sqrt{n}} < m < \bar{x} + 1.96\frac{\sigma}{\sqrt{n}}\right] = 0.95$$

1.6 Script

```
load 'plot'
NB. =====common=====
mean=: +/%#          NB. mean
dev=: -"1 mean      NB. deviation
var=:# %~ (+/ )@:*:@dev NB. Variance
cr=: % . } : "1     NB. cramer method
sel=:2 : ' m. {"1 n.' NB. select column (m.(column) sel n.(matrix))
NB. =====
machi_0=: 3 : 0
NB. y. is data(matrix vertical)// arrival time service time
NB. data type
M01=: y.,.^ . 1 sel y.      NB. add ln arrive time
LMDA=: % (0 { M02=: mean M01) NB. lambda
M04=: mean M01              NB. mean with ln
M05=: var M01               NB. var with ln
A=: +: ^ . 1{M04            NB. e^(m+(s^2)/2)
B=: ^ . (1{M05) + 1x1 ^ A   NB. e^(2m+2s^2)-e^(2m+s^2)
MAT=: 2 3 $ 2 ,1,A,2, 2 ,B NB. mat for cramer
M06=: cr MAT                NB. Theorem value
)

machi_00=: 3 : 0
NB. y. is mean and Var // arrival time service time
NB. mean (arrival) mean( service) //a b//
NB. var (arrival) var (service) //c d// dat=: 2 2 $ a b c d
```

```

M01=: y.
LMDA=: %(0 { { . y.)          NB. lambda
M04=: { . M01                  NB. mean with ln
M05=: {: M01                   NB. var with ln
A=: +: ^ . 1{M04               NB. e^(m+(s^2)/2)
B=: ^ . (1{M05) + 1x1 ^ A      NB. e^(2m+2s^2)-e^(2m+s^2)
MAT=: 2 3 $ 2 ,1,A,2, 2 ,B    NB. mat for cramer
M06=: cr MAT                   NB. Theorem value
)

machi_sim0=: 4 : 0
NB. x. nr. of customers (not need nr. of mashine)
NB. y. data (or machi_00 mean and var matrix)
NB. Usage: x. machi_sim0 y.
NB. calc arrival time and service time
NB. machi_0 y. NB. exchange machi_00
NB. M10=: rnd x.                NB. uniformed random
NB. random with mean and sd
NB. t = rno (mean 0 Var 1 nr. x.)// and e ^ m + sigma t
M10=: {: "1 M06
M11=: 1x1 ^S=: ({ . M10) + (%: {: M10)* rno 0,1,x.
NB. M11=: 1x1 ^S=: ({: M04) + (%: {: M05)* rno 0,1,x.
M13=: -(% LMDA)* ^ . M12=: rnd x.    NB. -1/lambda ln u
M12, .M13, .(>+/ L:0 <\ M13), .S, .M11
)

machi_sim1=: 4 : 0
NB. Usage: x.(e.g. 20 2) machi_sim1 y. (data)
X1=: { . x.                    NB. persons
X3=: (X2=:{: x.) # <0          NB. number of machine
NB. comment out next line //M15 is fixed and useful for analysys
NB. X1 machi_sim0 y.
NB. X1 y.

```

```

M15=(i. # tmp),. tmp,. M11,. (tmp=.>+/ L:0<\M13)+ M11
MACHINE=:/:~ ,{:("1) X2 { . M15 NB. input data to Mashine
COUNTER=: X2 NB. first step in Machine
SERV_START0=(1{"1) X2{ . M15)
SERV_START=:SERV_START0, ((# M15)-X2) # 0
T_MAT=: M15,.SERV_START,.((X2{ . 3{"1 M15),((#M15)-X2)#0) NB. main table
TINDEX0=:99999,.{:("1) X2{ . M15 NB. first step
TINDEX=:1 csort TMP=: (0 1{"1) M15),TINDEX0
  while. COUNTER < # M15 do.
    PRE=: 1 2 { COUNTER{T_MAT
    SV_TIME=: {: PRE
    if. (1{ COUNTER { T_MAT) > 0{MACHINE do.
      ST_TIME=: 1{COUNTER{T_MAT NB. ST_TIME is start service
      NB. calc
      END_TIME=: ST_TIME + SV_TIME
      T_MAT=(ST_TIME,END_TIME )(<COUNTER ;4 5 )}T_MAT
      MACHINE=:/:~( END_TIME (0) } MACHINE) NB. Machine newer
      TINDEX=: 1 csort TINDEX ,(99999,END_TIME)
    elseif. do.
      ST_TIME=: 0{MACHINE
      NB. WAIT=: (0{MACHINE)- ST_TIME

NB. same calc
      END_TIME=: ST_TIME + SV_TIME
      T_MAT=(ST_TIME,END_TIME )(<COUNTER ;4 5 )}T_MAT
      MACHINE=:/:~( END_TIME (0) } MACHINE) NB. Machine newer
      TINDEX=: 1 csort TINDEX ,(99999,END_TIME)
    end.
    NB. XXXX=: TINDEX ,. (1 2 3 {"1 IND exp_h M15),. neben_index TINDEX
    COUNTER=: >: COUNTER
  end.

T_MAT

```

)

```
machi_anal=: 4 : 0
NB. reading T_MAT and TINDEX//analysys
NB. x. y. is same as machi_sim1
NB. Usage: x.(a b) machi_anal y.(data)
NB. x. machi_sim1 y.
M20=: neben_tindex TINDEX
M21=: 1 4{"1 T_MAT
M22=(~/ "1 M21) # T_MAT NB. find not equal arrive and start
NB. wait is start service - arrive
NB. M23=: M22,. M24=: {: "1 (+/ ({"1 M22) =/ ( {"1 M20)) # M20
M23=: M22,. M24=: -/("1) 4 1 {"1 M22 NB. M24 is waiting time
)
```

```
wait_number=: 3 : 0
NB. count waiting number
NB. Usage: wait_number ''// '' is null
NB. this script should use same M21
NB. //so do machi_anal and soon continue wait_number
NB. count except 99999 and continue number is waiting persons
M25=: (=/"1 M21) # T_MAT NB. find not wait
M30 =: (tmp=:-.+/ ({"1 M25) =/ {"1 M20)# M20
)
```

```
count_number =: 3 : 0
NB. count waiting persons in line
NB. Usage: count_number '' //do soon until another mati_sim0
NB. M30=: wait_number ''
NB. not 99999 and cut by index/ drop head
NB. M31=:}. (1(0)} -. (L:0)99999= (L:0) {"1 M30}<;.1 {. "1 M30
NB. drop 99999 in each Box and count other in Box
IND=: 99999= {"1 M30
```



```

NB. =====Norman thomson=====
rnd=: (%~?)@(1e9&(#~)) NB. uniformed random number (0 1)
run=: +>(*rnd)/
rn=:--:~/@run@:(0 1&,)NB. Normal distribution
NB. Usage: rno a b c//a(mean) b(standard deviation c(number)
NB. e.g. nro 10 5 5
rno=: 3 : '({. y.)+(1{y.)*({: y.)(rn &>@#)12'

```

```

NB. =====

```

```

plot_random=: 3 : 0
NB. Usage e.g. plot_random 1000 bm 10000
Y=:>. /:~10 * y.
pd 'reset'
pd > #(L:0) (~: Y)<;.1 Y
pd 'show'
)

```

```

NB. =====
exp2=: 4 : 0
NB. expand and add 0 where flug is 0
NB. Usage: x. exp2 y.
NB. x. is flug
NB. y. is list to expand
index=: <: }.~. /:~ x. *>: i. # x.
y. index } ( # x.) # 0
)

```

```

NB. =====
NB. expand w & h like APL

```

```

NB. =====
exp_w=: 4 : 0
NB. expand Matrix data to Yoko(abreast)
NB. expand and add 0 where flug is 0
NB. Usage: x. exp2 y.
NB. x. is flug to expand ( binary)
NB. y. is Matrix or vector (auto-select) see select.
SIZE=: $ Y=: y.
INDEX=: <: }.~. /:~ x. *>: i. # x.
select. 1=#$ y. NB. select Vector or Matrix
case. 0 do. goto_matrix.
case. 1 do. goto_vector.
end.
label_matrix.
y. (<(i. # y.);INDEX) } tmp=:(({: SIZE),( # x.)) $ 0
return.
label_vector.
y. INDEX } ( # x.) # 0
return.
)

exp_h=: 4 : 0
NB. expand Matrix data to Tate(vertical)
NB. expand and add 0 where flug is 0
NB. Usage: x. exp2 y.
NB. x. is flug to expand ( binary)
NB. y. is Matrix for expand
SIZE=: $ Y=: y.
index=: <: }.~. /:~ x. *>: i. # x.
NB. y. (<(i. # y.);index) } tmp=:(({: SIZE),( # x.)) $ 0
y. (< index ; (i. {: $ y.)) } tmp=:(( # x.),({: SIZE)) $ 0
)

```

```
NB. =====DATA=====
NB. sample data dor machi_00 //To-hoku unuv. ATM data
tdata2=: 2 2 $ 59.54 97.78 4019.41 5895.53
```

1.6.1 reference

東北大学統計学グループ「これだけは知っておこう統計学」有斐閣 2002

Norman Thomson [J: the Natural Language for Analytic Computing] Resuarch Studies
Press 2001