Queuing Theory Cookbook

Contents

```
Arrivals ----λ----> Queue --------> Service ----μ----> Done
               from                   First in
       unlimited           is
      population         first out
                                   (FIFO)
```

Steady state results for single-server single-stage queue with
Poisson arrivals. λ arrive per unit of time.
Exponential service times. μ served per unit of time.
First-come first-served queue discipline.
Unlimited length queue.
Drawing on an Unlimited population (so the future arrival rate does not depend on the past arrival rate).

Server utilization factor
(what proportion of the time the server is busy) = $\rho = \lambda/\mu$

Probability of 0 in system $$Prob(0) = 1 - \rho$$

Probability of n in system $$Prob(n) = \rho^n(1-\rho)$$

Average number of customers in system $$L = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$$

Average number in queue $$L_q = L-\rho = \frac{\rho^2}{1-\rho}$$

Average time in system $$W = \frac{L}{\lambda} = \frac{1}{\mu-\lambda}$$

Average wait in queue $$W_q = \frac{L_q}{\lambda} = \frac{\rho}{\mu-\lambda}$$

Spreadsheet for single stage, single-server system. Poisson arrivals (λ) and service completions (μ), FIFO queue discipline, unlimited length queue, drawing on an unlimited population.
In this example, λ=3 and μ=4.

	A	B	C	D	E
1	Single-ser				
2	Lambda	3		L	=B4/(1-B4)
3	Mu	4		Lq	=E2-B4
4	Rho	=B2/B3		W	=E2/B2
5				Wq	=E3/B2
6					
7	n	P(n)	P(<=n)	P(>n)	
8	0	=1-B4	=B8	=1-C8	
9	=A8+1	=$B$4^A9*(1-$B$4)	=C8+B9	=1-C9	
10	=A9+1	=$B$4^A10*(1-$B$4)	=C9+B10	=1-C10	
11	=A10+1	=$B$4^A11*(1-$B$4)	=C10+B11	=1-C11	
12	=A11+1	=$B$4^A12*(1-$B$4)	=C11+B12	=1-C12	
13	=A12+1	=$B$4^A13*(1-$B$4)	=C12+B13	=1-C13	
14	=A13+1	=$B$4^A14*(1-$B$4)	=C13+B14	=1-C14	

To make the table of probabilities, type the formulas in cells A9:D9. Then copy and paste them to A9:A14 (or as far down as you want.)

	A	B	C	D	E
1	Single-server, single stage				
2	Lambda	3		L	3
3	Mu	4		Lq	2.25
4	Rho	0.75		W	1
5				Wq	0.75
6					
7	n	P(n)	P(<=n)	P(>n)	
8	0	0.25	0.25	0.75	
9	1	0.1875	0.4375	0.5625	
10	2	0.140625	0.578125	0.421875	
11	3	0.1054688	0.6835938	0.3164063	
12	4	0.0791016	0.7626953	0.2373047	
13	5	0.0593262	0.8220215	0.1779785	
14	6	0.0444946	0.8665161	0.1334839	

If you use the formulas shown above in your own spreadsheet, use this picture to check your formulas. Put 3 in cell B2 for Lambda and 4 in B3 for Mu, as shown here. Then see if your other numbers match the other numbers here.

```
Arrivals ----λ----> Queue --------> Service ----μ----> Done
               from                 First in
             Limited              is
            population          first out
            of size N           (FIFO)
```

Steady state results for single-server single-stage queue with
Drawing on an Limited population of size N.
λ is how often each potential customer comes in on average, when not already in the queue or being served.
Total arrivals per unit time is between λ and Nλ.
Exponential service time. μ served per unit of time.
First-come first-served queue discipline.
Unlimited length queue.

We use $\rho = \lambda/\mu$
in the equations,
but it is not the proportion of the time the server is busy in this case.

Probability of 0 in system

$$Prob(0) = \frac{1}{\sum_{i=0}^{N} \frac{N!}{(N-i)!}\rho^i}$$

Probability of n in system

$$Prob(n) = Prob(0)\frac{N!}{(N-n)!}\rho^n$$

Average number of customers in system

$$L = N - \frac{1 - Prob(0)}{\rho}$$

Average number in queue

$$L_q = L - (1 - Prob(0))$$

Average time in system

$$W = \frac{L}{\lambda(N-L)}$$

Average wait in queue

$$W_q = \frac{L_q}{\lambda(N-L)}$$

Example: A maintenance department is responsible for 6 monitoring machines. Each machine breaks down an average of once per 7 days. Fixing the machines takes two days on average.

	A	B	C
1	Single-ser		
2	Lambda	=1/7	
3	Mu	=1/2	
4	Rho	=B2/B3	
5	N	=COUNT(A9:A100)-1	
6			
7	To get pro		
8	I or n	(N!/(N-i)!)Rho^I	
9	0	1	
10	=A9+1	=FACT($B$5)/FACT($B$5-A10)*$B$4^A10	
11	=A10+1	=FACT($B$5)/FACT($B$5-A11)*$B$4^A11	
12	=A11+1	=FACT($B$5)/FACT($B$5-A12)*$B$4^A12	
13	=A12+1	=FACT($B$5)/FACT($B$5-A13)*$B$4^A13	
14	=A13+1	=FACT($B$5)/FACT($B$5-A14)*$B$4^A14	
15	=A14+1	=FACT($B$5)/FACT($B$5-A15)*$B$4^A15	

The formula in D9 for Prob(0) is implemented using column B. B9 on down has the individual terms in the sum in the denominator of the Prob(0) formula. D9's formula includes a sum from B9 to B100. Also, N (in B5) is not hard-coded, but is calculated by counting the number of filled-in rows starting with row 9. Together, these allow you to change N by adding or taking out rows starting after row 10. All rows after 10 are copies of row 10.

	D	E
1		
2	L	=B5-(1-D9)/B4
3	Lq	=E2-(1-D9)
4	W	=E2/(B2*(B5-E2))
5	Wq	=E3/(B2*(B5-E2))
6		
7		
8	Prob(n in system)	
9	=1/SUM(B9:B100)	
10	=$D$9*FACT($B$5)/FACT($B$5-A10)*$B$4^A10	
11	=$D$9*FACT($B$5)/FACT($B$5-A11)*$B$4^A11	
12	=$D$9*FACT($B$5)/FACT($B$5-A12)*$B$4^A12	
13	=$D$9*FACT($B$5)/FACT($B$5-A13)*$B$4^A13	
14	=$D$9*FACT($B$5)/FACT($B$5-A14)*$B$4^A14	
15	=$D$9*FACT($B$5)/FACT($B$5-A15)*$B$4^A15	

L (see below) is about 2.8, meaning that, on average, 2.8 machines are being fixed or waiting to be fixed. This implies that, of your 6 machines, only 3.2 are working at any given time, on average.

	A	B	C	D	E
1	Single-server, single stage, limited population of N				
2	Lambda	0.142857		L	2.788692
3	Mu	0.5		Lq	1.871176
4	Rho	0.285714		W	6.078784
5	N	6		Wq	4.078784
6					
7	To get prob(0 in system):				
8	I or n	(N!/(N-i)!)Rho^I		Prob(n in system)	
9	0	1		0.082484	
10	1	1.714286		0.1414	
11	2	2.44898		0.202001	
12	3	2.798834		0.230858	
13	4	2.399		0.197878	
14	5	1.370857		0.113073	
15	6	0.391674		0.032307	

Suppose you need to have 5 machines working at once, on average. You can use this type of analysis to help figure out whether you should buy more machines or speed up your maintenance department. You would also need to know the cost of a machine and the cost of speeding up service.

```
Arrivals ----λ----> Queue --------> Service ----μ----> Done
             from                   First in
                    Unlimited              is
             population             first out (FIFO)
```

Queue limited to c-1
Maximum of c in system

Steady state results for <u>single-server single-stage queue</u> with
Poisson arrivals. λ arrive per unit of time.
Exponential service times. μ served per unit of time.
First-come first-served queue discipline.
Limited size of system: No more that c allowed in system.
Drawing on an Unlimited population (so the future arrival rate does not depend on the past arrival rate).

We use $\rho = \lambda/\mu$
in the equations,
but it is not the proportion of the time the server is busy in this limited-size model. The server is busy less than ρ. There can be a steady state even if $\rho > 1$.

Probability of 0 in system

$$Prob(0) = \frac{1-\rho}{1-\rho^{c+1}}$$

Probability of n in system

$$Prob(n) = Prob(0)\rho^n$$

Average number in system

$$L = \frac{\rho}{1-\rho} - \frac{(c+1)\rho^{c+1}}{1-\rho^{c+1}}$$

Average length of queue

$$L_q = L - \rho$$

Average time in system

$$W = \frac{L}{\lambda}$$

Average wait in queue

$$W_q = \frac{L_q}{\lambda}$$

Single server single stage system with truncated queue. No more than c may be in the system. This means no more than c-1 can be in the queue.

	A	B	C	D
1	Lambda	3		
2	Mu	4		
3	Rho	=B1/B2		
4	c	3		
5				
6	L	=B3/(1-B3)-(B4+1)*B3^(B4+1)/(1-B3^(B4+1))		
7	Lq	=B6-B3		
8	W	=B6/B1		
9	Wq	=B7/B1		
10				
11	n	Prob(n)	Prob(<=n)	Prob(>n)
12	0	=(1-B3)/(1-B3^(B4+1))	=B12	=1-C12
13	=A12+1	=$B$12*$B$3^A13	=C12+B13	=1-C13
14	=A13+1	=$B$12*$B$3^A14	=C13+B14	=1-C14
15	=A14+1	=$B$12*$B$3^A15	=C14+B15	=1-C15

This example has c = 3. This corresponds to a maximum queue length of 2, since there is one server, and 2+1=3. For higher values of c, copy and paste row 13 down as far as needed.

	A	B	C	D
1	Lambda	3		
2	Mu	4		
3	Rho	0.75		
4	c	3		
5				
6	L	1.1485714		
7	Lq	0.3985714		
8	W	0.3828571		
9	Wq	0.1328571		
10				
11	n	Prob(n)	Prob(<=n)	Prob(>n)
12	0	0.3657143	0.3657143	0.6342857
13	1	0.2742857	0.64	0.36
14	2	0.2057143	0.8457143	0.1542857
15	3	0.1542857	1	0

You need only enough rows to get to n equal to c. In this example, with c=3, we only need to go down to where n is 3.

In this example, with the system limited to 3, the probability of 3 or less in the system is 1, indicating certainty, and the probability of more than 3 in the system is 0, indicating that there is no chance of that happening.

```
Arrivals ---->λ----> Queue --------> Service ----μ----> Done
      from                First in        Constant
       Unlimited           is              Service
         population          first out       Time
                                    (FIFO)
```

Steady state results for <u>single-server single-stage queue</u> with
Poisson arrivals. λ arrive per unit of time.
Constant service time. μ served per unit of time. Every service takes exactly 1/μ amount of time.
First-come first-served queue discipline.
Unlimited length queue.
Drawing on an Unlimited population (so the future arrival rate does not depend on the past arrival rate).

Server utilization factor
(what proportion of the time the server is busy)= $\rho = \lambda/\mu$

Probability of 0 in system $$Prob(0) = 1-\rho$$

Probability of 1 in system $$Prob(1) = (1-\rho)(e^{\rho}-1)$$

Probability of n in system (n ≥ 2) $$(1-\rho)\left[\sum_{i=1}^{n}(-1)^{n-i}\frac{(\rho i)^{n-i}}{(n-i)!} + \sum_{i=1}^{n-1}e^{\rho i}(-1)^{n-i}\frac{(\rho i)^{n-i}}{(n-1-i)!}\right]$$

Average number in system $$L = \rho+\frac{\rho^2}{2(1-\rho)}$$

Average number in queue $$L_q = L - \rho$$

Average time in system $$W = \frac{L}{\lambda}$$

Average wait in queue $$W_q = \frac{L_q}{\lambda} = W-\frac{1}{\mu}$$

Single server system with constant service time.

I did not attempt the Prob(n) formula for n > 1.

	A	B
1	Constant s	
2	Lambda	3
3	Mu	4
4	Rho	=B2/B3
5		
6	Prob(0)	=1-B4
7		
8	L	=B4+B4^2/(2*(1-B4))
9	Lq	=B8-B4
10	W	=B8/B2
11	Wq	=B10-1/B3

The numbers, for λ=3 and μ=4.

	A	B
1	Constant service time	
2	Lambda	3
3	Mu	4
4	Rho	0.75
5		
6	Prob(0)	0.25
7		
8	L	1.875
9	Lq	1.125
10	W	0.625
11	Wq	0.375

```
Arrivals ⟶ Queue ⟶ Service ⟶ Service ⟶ ... ⟶ Service ⟶ Done
                         k stages of service
```

Erlang Distribution

All stages have to be completed for one customer before the next customer can be served. A telephone call would be an example. Erlang worked for the Danish phone company in the early 1900's.

Steady state results for <u>single-server multi-stage queue</u> with
Poisson arrivals. λ arrive per unit of time.
Exponential distribution service times for each stage.
μ served per unit of time for all stages total.
k is the number of stages, so each at each stage average service time is 1/(kμ).
First-come first-served queue discipline.
Unlimited length queue.
Drawing on an Unlimited population, so the future arrival rate does not depend on the past arrival rate.

Server utilization factor (what proportion of the time the server -- at least one stage -- is busy) = ρ = λ/μ

Probability of 0 in system Prob(0) = 1 - ρ

Probability of n in system (General formula too messy)

Average number in system $$L = \rho + \frac{(k+1)\rho^2}{2k(1-\rho)}$$

Average number in queue $$L_q = L - \rho$$

Average time in system $$W = \frac{L}{\lambda}$$

Average wait in queue $$W_q = \frac{L_q}{\lambda} = W - \frac{1}{\mu}$$

Example of the Erlang distribution: Steady state results for single-server multi-stage queue with Poisson arrivals. λ arrive per unit of time. Exponential distribution service times for each stage. μ served per unit of time for all stages total. k is the number of stages. Each stage is assumed to take the same time, $1/(k\mu)$, on average.

In this example, there are three stages, so k=3.

	A	B
1	Erlang	
2	Lambda	3
3	Mu	4
4	rho	=B2/B3
5	k stages	3
6		
7	Prob(0)	=1-B4
8		
9	L	=B4+(B5+1)*B4^2/(2*B5*(1-B4))
10	Lq	=B9-B4
11	W	=B9/B2
12	Wq	=B11-1/B3

	A	B
1	Lambda	3
2	Mu	4
3	rho	0.75
4	k stages	3
5		
6	Prob(0)	0.25
7		
8	L	2.25
9	Lq	1.5
10	W	0.75
11	Wq	0.5

```
                            +--> Service --μ--+
Arrivals ---λ---> Queue --+--> Service --μ--+-->Done
                            +--> Service --μ--+
                               M servers in parallel
```

Steady state results for a Multiple-Server Single-Stage Queue, meaning that we have one line that leads to several servers, each of whom can serve any customer equally well.

Poisson arrivals. λ arrive per unit of time.
M servers working in parallel.
Exponential service times. μ served per unit of time by each server.
First-come first-served queue discipline.
Unlimited length queue.
Drawing on an Unlimited population (so the future arrival rate does not depend on the past arrival rate).

Server utilization factor (what proportion of the time an individual server is busy) = $\rho = \lambda/(M\mu)$

Probability of 0 in system

$$Prob(0) = \frac{1}{\sum_{i=0}^{M-1} \frac{(\rho M)^i}{i!} + \frac{(\rho M)^M}{M!(1-\rho)}}$$

Probability of n in system

$$Prob(n) = Prob(0)\frac{(\rho M)^n}{n!} \quad if\ n \le M$$

$$Prob(n) = Prob(0)\frac{\rho^n M^M}{M!} \quad if\ n \ge M$$

Average number in system

$$L = L_q + \frac{\lambda}{\mu}$$

Average number in queue

$$L_q = Prob(0)\frac{M^M \rho^{M+1}}{M!(1-\rho)^2}$$

Average time in system

$$W = \frac{L}{\lambda}$$

Average wait in queue

$$W_q = \frac{L_q}{\lambda} = W - \frac{1}{\mu}$$

Spreadsheet layout for two server, single stage system.

	A	B	C	D	E
1	Two servers				
2	Lambda	3		L	=E3+B2/B3
3	Mu	4		Lq	=B8*4*B5^3/(2*(1-B5)^2)
4	M	2		W	=E2/B2
5	rho	=B2/(B3*2)		Wq	=E3/B2
6					
7	n	Prob(n)			
8	0	=1/(1+B5*2+(B5*2)^2/(2*(1-B5)))			
9	1	=B8*B5*2			
10	=A9+1	=$B$8*$B$5^A10*2			
11	=A10+1	=$B$8*$B$5^A11*2			
12	=A11+1	=$B$8*$B$5^A12*2			
13	=A12+1	=$B$8*$B$5^A13*2			

This spreadsheet only works for two servers. It uses the number 2 wherever the formulas call for M. This means that changing M in this spreadsheet won't change the calculated results.

Rows after 10 can be created by copying and pasting from row 10.

	A	B	C	D	E
1	Two servers in parallel				
2	Lambda	3		L	0.8727273
3	Mu	4		Lq	0.1227273
4	M	2		W	0.2909091
5	rho	0.375		Wq	0.0409091
6					
7	n	Prob(n)			
8	0	0.4545455			
9	1	0.3409091			
10	2	0.1278409			
11	3	0.0479403			
12	4	0.0179776			
13	5	0.0067416			

Multiple server, single stage system. For this spreadsheet.you can use any value for M. The number of filled-in rows after row 13 must be greater than or equal to M.

Rows after 14 are copies of row 14. The formula for the probability of 0 involves a sum that goes down to B105. The 105 number was arbitrarily picked to be larger than any M you were likely to use.

Formulas in the B and C columns use the IF function. The general form of the IF function is:
IF(this statement is true, then use this, otherwise use this).

	A	B
1	M servers	
2	Lambda	3
3	Mu	4
4	M	3
5	Rho	=B2/(B4*B3)
6		
7	L	=B8+B2/B3
8	Lq	=C13*B4^B4*B5^(B4+1)/(FACT(B4)*(1-B5)^2)
9	W	=B7/B2
10	Wq	=B8/B2
11		
12	i or n	(RhoM)^I/I!
13	0	1
14	=A13+1	=IF(A14<$B$4,($B$5*$B$4)^A14/FACT(A14),0)
15	=A14+1	=IF(A15<$B$4,($B$5*$B$4)^A15/FACT(A15),0)
16	=A15+1	=IF(A16<$B$4,($B$5*$B$4)^A16/FACT(A16),0)
17	=A16+1	=IF(A17<$B$4,($B$5*$B$4)^A17/FACT(A17),0)
18	=A17+1	=IF(A18<$B$4,($B$5*$B$4)^A18/FACT(A18),0)
19	=A18+1	=IF(A19<$B$4,($B$5*$B$4)^A19/FACT(A19),0)
20	=A19+1	=IF(A20<$B$4,($B$5*$B$4)^A20/FACT(A20),0)
21	=A20+1	=IF(A21<$B$4,($B$5*$B$4)^A21/FACT(A21),0)

	C
12	Prob(n)
13	=1/(SUM(B13:B105)+(B5*B4)^B4/(FACT(B4)*(1-B5)))
14	=$C$13*$B$5^A14*(IF(A14<$B$4,$B$4^A14/FACT(A14),$B$4^$B$4/FACT($B$4)))
15	=$C$13*$B$5^A15*(IF(A15<$B$4,$B$4^A15/FACT(A15),$B$4^$B$4/FACT($B$4)))
16	=$C$13*$B$5^A16*(IF(A16<$B$4,$B$4^A16/FACT(A16),$B$4^$B$4/FACT($B$4)))
17	=$C$13*$B$5^A17*(IF(A17<$B$4,$B$4^A17/FACT(A17),$B$4^$B$4/FACT($B$4)))
18	=$C$13*$B$5^A18*(IF(A18<$B$4,$B$4^A18/FACT(A18),$B$4^$B$4/FACT($B$4)))
19	=$C$13*$B$5^A19*(IF(A19<$B$4,$B$4^A19/FACT(A19),$B$4^$B$4/FACT($B$4)))
20	=$C$13*$B$5^A20*(IF(A20<$B$4,$B$4^A20/FACT(A20),$B$4^$B$4/FACT($B$4)))
21	=$C$13*$B$5^A21*(IF(A21<$B$4,$B$4^A21/FACT(A21),$B$4^$B$4/FACT($B$4)))

The expression in B16, for example, puts the value of (B5*B4)^A16/FACT(A16) in this cell if A16 is less than B4. If A16 is bigger than or equal to B4, the expression puts 0 in this cell.

Here's what that spreadsheet looks like for M=3.

	A	B	C
1	M servers in parallel		
2	Lambda	3	
3	Mu	4	
4	M	3	
5	Rho	0.25	
6			
7	L	0.764706	
8	Lq	0.014706	
9	W	0.254902	
10	Wq	0.004902	
11			
12	i or n	(RhoM)^I/I	Prob(n)
13	0	1	0.470588
14	1	0.75	0.352941
15	2	0.28125	0.132353
16	3	0	0.033088
17	4	0	0.008272
18	5	0	0.002068
19	6	0	0.000517
20	7	0	0.000129
21	8	0	3.23E-05

With these arrival and service rates, 3 servers just about eliminates waiting. Wq is under 18 seconds. On the other hand, your servers are busy only one-fourth of the time.

```
                    +---> Service ┐
         Arrivals ---+---> Service ┼─▸ Done
                    +---> Service ┘
                       M servers in parallel
```

No queue allowed (absolute truncation – nobody waits)

Average length of stay in service = 1/μ

Steady state results for a <u>Multiple-Server Single-Stage Queue with absolute truncation</u>. This means that there are several servers but no waiting. Hotel rooms and some health care functions (e.g. maternity) are examples.

Poisson arrivals. λ arrive per unit of time.
M servers working in parallel. Exponential service times. μ served per unit of time by each server.
No queue.
Drawing on an Unlimited population.

Server utilization factor (what proportion of the time an individual server is busy) = $\rho = \lambda/(M\mu)$

Probability of n in system

$$Prob(n) = \frac{\dfrac{(\frac{\lambda}{\mu})^n}{n!}}{\sum_{i=0}^{M} \dfrac{(\frac{\lambda}{\mu})^i}{i!}}$$

Average number in system

$$L = (\frac{\lambda}{\mu}) \frac{\sum_{i=0}^{M-1} \dfrac{(\frac{\lambda}{\mu})^i}{i!}}{\sum_{i=0}^{M} \dfrac{(\frac{\lambda}{\mu})^i}{i!}}$$

L_q and W_q are not defined because no queue is allowed.

W is the service time, $1/\mu$.

Multiple server single stage queue with queue truncation at 0.
M is the number of servers. Lambda is the average number of arrivals per unit of time. Arrivals that can't be served are turned away. Mu is the average number of customers each server serves per unit of time.

In this example, imagine a rehab facility with 8 beds. (M=8.) Sixteen patients arrive per month. (λ = 16). The average patient stays for a week. ($1/\mu$ = 1/4, μ = 4. A week is 1/4 of a month.)

	A	B	C
1	Lambda	16	
2	Mu	4	
3	M	8	
4	Rho	=B5/B3	
5	M*Rho	=B1/B2	= lambda/mu
6			
7	Sums		=SUM(C10:C100)
8			
9	I or n	(lambda/mu)^I/I!	n <= M-1
10	0	1	=IF(A10<$B$3,B10,0)
11	=A10+1	=$B$5^A11/FACT(A11)	=IF(A11<$B$3,B11,0)
12	=A11+1	=$B$5^A12/FACT(A12)	=IF(A12<$B$3,B12,0)
13	=A12+1	=$B$5^A13/FACT(A13)	=IF(A13<$B$3,B13,0)
14	=A13+1	=$B$5^A14/FACT(A14)	=IF(A14<$B$3,B14,0)
15	=A14+1	=$B$5^A15/FACT(A15)	=IF(A15<$B$3,B15,0)
16	=A15+1	=$B$5^A16/FACT(A16)	=IF(A16<$B$3,B16,0)
17	=A16+1	=$B$5^A17/FACT(A17)	=IF(A17<$B$3,B17,0)
18	=A17+1	=$B$5^A18/FACT(A18)	=IF(A18<$B$3,B18,0)
19	=A18+1	=$B$5^A19/FACT(A19)	=IF(A19<$B$3,B19,0)
20	=A19+1	=$B$5^A20/FACT(A20)	=IF(A20<$B$3,B20,0)

	D	E
1	L	=B5*C7/D7
2	W	=1/B2
3		
4		
5		
6		
7	=SUM(D10:D100)	
8		
9	n <= M	Prob(n)
10	=IF(A10<=$B$3,B10,0)	=IF(A10<=$B$3,B10/$D$7,0)
11	=IF(A11<=$B$3,B11,0)	=IF(A11<=$B$3,B11/$D$7,0)
12	=IF(A12<=$B$3,B12,0)	=IF(A12<=$B$3,B12/$D$7,0)
13	=IF(A13<=$B$3,B13,0)	=IF(A13<=$B$3,B13/$D$7,0)
14	=IF(A14<=$B$3,B14,0)	=IF(A14<=$B$3,B14/$D$7,0)
15	=IF(A15<=$B$3,B15,0)	=IF(A15<=$B$3,B15/$D$7,0)
16	=IF(A16<=$B$3,B16,0)	=IF(A16<=$B$3,B16/$D$7,0)
17	=IF(A17<=$B$3,B17,0)	=IF(A17<=$B$3,B17/$D$7,0)
18	=IF(A18<=$B$3,B18,0)	=IF(A18<=$B$3,B18/$D$7,0)
19	=IF(A19<=$B$3,B19,0)	=IF(A19<=$B$3,B19/$D$7,0)
20	=IF(A20<=$B$3,B20,0)	=IF(A20<=$B$3,B20/$D$7,0)

	A	B	C	D	E
1	Lambda	16		L	3.8783
2	Mu	4		W	0.25
3	M	8			
4	Rho	0.5			
5	M*Rho	4	= lambda/mu		
6					
7	Sums		51.80635	53.432	
8					
9	I or n	(lambda/m	n <= M-1	n <= M	Prob(n)
10	0	1	1	1	0.0187
11	1	4	4	4	0.0749
12	2	8	8	8	0.1497
13	3	10.66667	10.66667	10.667	0.1996
14	4	10.66667	10.66667	10.667	0.1996
15	5	8.533333	8.533333	8.5333	0.1597
16	6	5.688889	5.688889	5.6889	0.1065
17	7	3.250794	3.250794	3.2508	0.0608
18	8	1.625397	0	1.6254	0.0304
19	9	0.722399	0	0	0
20	10	0.288959	0	0	0
21					

The average number of patients in the facility is L, 3.878, which means the occupancy rate is 3.878/8 = 48%. The facility is full (Prob(8)) 3% of the time, so you take in 97% of the patients referred to you.

By trying different numbers for M you can see what happens if you have more or fewer beds. The results can be surprising. For example, if you cut M to 4, your occupancy rate rises only to 69%, because your average census (L) drops to 2.76. This is because you're now full 31% of the time, so you're taking in only 69% of the patients who come to you.

```
               ┌─▸ Queue 1 ─┬─┐  ┌─▸ Service         ┐
Arrivals          ──┼─▸ Queue 2 ─┤  └─┼─▸ Service ─┼─▸ Bye-Bye
               └─▸ Queue 3 ─┘        └─▸ Service ─┘
                   G priority classes   M servers in parallel
```

Customers in lower number priority classes go before customers with higher numbers. Within priority classes, service is first-in-first-out.

"Preemptive" versus "Non-preemptive" priorities has to do with what happens if an urgent customer comes in while a less urgent customer is being served:

Preemptive priorities: Service to a customer is interrupted if another customer arrives who is in a lower priority class.

Non-preemptive priorities: Once a customer's service is started, the customer is not bumped if a lower priority class customer arrives.

Poisson arrivals. λ_g arrive per unit of time in priority class g. Total arrivals, λ, is the sum of the λ_g's.
Exponential service times. μ served per unit of time by each server.
Unlimited length queue. Drawing on an unlimited population.

Steady state results:

If priorities are non-preemptive, calculate A :

$$If\ M = 1\ then\quad A = \frac{\mu^2}{\lambda}$$

$$If\ M > 1\ then\quad A = M\mu + \frac{M!(M\mu-\lambda)}{(\frac{\lambda}{\mu})^M}\sum_{i=1}^{M-1}\frac{(\frac{\lambda}{\mu})^i}{i!}$$

For preemptive or non-preemptive priorities, calculate B_0 to B_G using these formulas:
(G is the number of groups. In the formulas below, g is the group number.)

$$B_0 = 1$$

$$B_g = 1 - \frac{1}{M\mu}\sum_{i=1}^{g}\lambda_i$$

For non-preemptive priorities, the average time in spent in the system and in queue for priority class g is:

$$W_g = \frac{1}{AB_{g-1}B_g} + \frac{1}{\mu} \qquad W_{gq} = \frac{1}{AB_{g-1}B_g}$$

For preemptive priorities, the average time spent in the system and in the queue for priority class g is:

$$W_g = \frac{1}{\mu B_{g-1} B_g} \qquad W_{gq} = W_g - \frac{1}{\mu}$$

For both preemptive and non-preemptive queues, the average number of persons in system and in queue for priority class g is:

$$L_g = \lambda_g W_g \qquad L_{gq} = \lambda_g W_{gq}$$

Spreadsheet for queue with preemptive priorities, three priority classes, one server.

Notice the use of $ signs in the formulas to take advantage of relative and absolute references. Cells B22:C24 were filled in by creating the entry in B22, cutting it, and then pasting it to B22:C24.

	A	B	C	D
1	Preemptive p			
2	An emergen			
3	Mu	4	per hour	
4	People arrive			
5			per hour, consisting of	
6	Lambda1	0.5	emergencies	
7	Lambda2	1	serious illnesses	
8	Lambda3	1.5	others	
9				
10	Intermediate			
11	B0	1		
12	B1	=1-SUM($B$6:B6)/$B$3		
13	B2	=1-SUM($B$6:B7)/$B$3		
14	B3	=1-SUM($B$6:B8)/$B$3		
15				
16	Time	Waiting Wq	System W	
17	W1	=C17-1/$B$3	=1/($B$3*B12*B11)	hours
18	W2	=C18-1/$B$3	=1/($B$3*B13*B12)	hours
19	W3	=C19-1/$B$3	=1/($B$3*B14*B13)	hours
20				
21	Number	Waiting Lq	in System L	
22	L1	=B17*$B6	=C17*$B6	people
23	L2	=B18*$B7	=C18*$B7	people
24	L3	=B19*$B8	=C19*$B8	people

At these arrival and service rates, which total the same λ=3 and μ=4 as on earlier examples, waiting time is very different for the different priority classes. Emergencies get seen on average in .036 hours, which is 2.14 minutes. "Others" wait an average of 1.35 hours, which is about 96 minutes. The emergency room waiting area will average having two low-priority people in it.

	A	B	C	D	E
1	Preemptive priorities				
2	An emergency room handles people at the rate of				
3	Mu	4	per hour		
4	People arrive at				
5			per hour, consisting of		
6	Lambda1	0.5	emergencies		
7	Lambda2	1	serious illnesses		
8	Lambda3	1.5	others		
9					
10	Intermediate results:				
11	B0	1			
12	B1	0.875			
13	B2	0.625			
14	B3	0.25			
15					
16	Time thru	Waiting Wq	System W		
17	W1	0.035714	0.285714	hours	
18	W2	0.207143	0.457143	hours	
19	W3	1.35	1.6	hours	
20					
21	Number	Waiting	in System		
22	L1	0.017857	0.142857	people	
23	L2	0.207143	0.457143	people	
24	L3	2.025	2.4	people	

Spreadsheet for queue with non-preemptive priorities, three priority classes, one server.

	A	B	C	D
1	Non-preemptive			
2	An emergency r			
3	Mu	4	per hour	
4	People arrive at			
5			per hour, consisti	
6	Lambda1	0.5	emergencies	
7	Lambda2	1	serious illnesses	
8	Lambda3	1.5	others	
9				
10	Intermediate res			
11	A	=B3^2/SUM(B6:B8)		
12	B0	1		
13	B1	=1-SUM($B$6:B6)/$B$3		
14	B2	=1-SUM($B$6:B7)/$B$3		
15	B3	=1-SUM($B$6:B8)/$B$3		
16				
17	Time thru	Waiting Wq	System W	
18	W1	=1/($B$11*B13*B12)	=B18+1/$B$3	hours
19	W2	=1/($B$11*B14*B13)	=B19+1/$B$3	hours
20	W3	=1/($B$11*B15*B14)	=B20+1/$B$3	hours
21				
22	Number	Waiting	in System	
23	L1	=B18*$B6	=C18*$B6	people
24	L2	=B19*$B7	=C19*$B7	people
25	L3	=B20*$B8	=C20*$B8	people

Non-preemptive priorities makes the wait longer for emergencies than does preemptive priorities. Least-urgent patients wait less.

	A	B	C	D	E
1	Non-preemptive priorities				
2	An emergency room handles people at the rate of				
3	Mu	4	per hour		
4	People arrive at				
5			per hour, consisting of		
6	Lambda1	0.5	emergencies		
7	Lambda2	1	serious illnesses		
8	Lambda3	1.5	others		
9					
10	Intermediate results:				
11	A	5.333333			
12	B0	1			
13	B1	0.875			
14	B2	0.625			
15	B3	0.25			
16					
17	Time thru	Waiting W	System W		
18	W1	0.214286	0.464286	hours	
19	W2	0.342857	0.592857	hours	
20	W3	1.2	1.45	hours	
21					
22	Number	Waiting	in System		
23	L1	0.107143	0.232143	people	
24	L2	0.342857	0.592857	people	
25	L3	1.8	2.175	people	